

令和元年9月19日判決言渡

平成31年(行ケ)第10005号 審決取消請求事件

口頭弁論終結日 令和元年7月4日

判 決

原 告 株式会社三菱UFJ銀行

同訴訟代理人弁護士 高 橋 雄 一 郎

阿 部 実 佑 季

同訴訟代理人弁理士 林 佳 輔

福 永 健 司

被 告 特許庁長官

同 指 定 代 理 人 須 田 勝 巳

辻 本 泰 隆

松 平 英

野 崎 大 進

豊 田 純 一

主 文

- 1 特許庁が不服2018-3406号事件について平成30年11月30日にした審決を取り消す。
- 2 訴訟費用は被告の負担とする。

事 実 及 び 理 由

第1 請求

主文同旨

第2 事案の概要

本件は、特許出願拒絶査定に対する不服審判請求を不成立とした審決の取消訴訟である。争点は、独立特許要件違反（進歩性欠如）の判断の誤りの有無である。

1 特許庁における手続の経緯

原告は、発明の名称を「アプリケーション生成支援システムおよびアプリケーション生成支援プログラム」とする発明につき、平成29年6月26日、特許出願（特願2017-124385号。請求項の数34。甲9の1～3。以下「本願」という。）をしたが、平成30年1月17日付けで拒絶査定（甲13）を受けたので、同年3月8日、拒絶査定不服審判請求をし、同審判請求は、不服2018-3406号として審理された（甲14）。

原告は、同年6月19日、特許請求の範囲を補正する手続補正（請求項の数16。甲18）をし、その後、同年9月12日、特許請求の範囲を補正する手続補正（請求項の数2。甲21。以下「本件補正」という。）をしたが、特許庁は、同年11月30日、本件補正を却下した上、「本件審判の請求は、成り立たない。」との審決（以下「本件審決」という。）をし、同審決謄本は、同年12月18日、原告に送達された。

2 特許請求の範囲の記載（甲18，21）

(1) 本件補正前の本願の特許請求の範囲請求項1の記載は、次のとおりである（同請求項に係る発明を、以下「本件補正前発明」という。）。

「通信端末に固有のネイティブ機能を実行させるためのパラメータに応じて、前記通信端末において実行されるアプリケーションの動作を規定する設定ファイルを設定する設定部と、

前記設定ファイルに基づいてアプリケーションパッケージを生成する生成部と、を有するアプリケーション生成支援システム。」

(2) 本件補正後の本願の特許請求の範囲請求項1の記載は、次のとおりである（同請求項に係る発明を、以下「本件補正発明」という。）。

「携帯通信端末に固有のネイティブ機能を実行させるためのパラメータに応じて、

前記携帯通信端末において実行されるアプリケーションの、前記携帯通信端末の動きに伴う動作を規定する設定ファイルを設定する設定部と、

前記設定ファイルに基づいてアプリケーションパッケージを生成する生成部と、を有するアプリケーション生成支援システム。」

3 本件審決の理由の要旨

(1) 引用発明及び周知技術について

ア 主引例

特許第5470500号公報（平成26年4月16日発行。以下「引用文献1」という。）には、以下のとおりの発明（以下「引用発明」という。）が記載されている。

「アプリケーション生成装置1と開発用端末2とがインターネット等のネットワークNを介して通信可能に接続され、スマートフォン等の携帯端末にインストールされるアプリケーションプログラムであるネイティブアプリケーションを生成するアプリケーション生成システムであって、

アプリケーション生成装置1は、記憶部11と、受付部12と、生成部13と、変換部14と、送信部15とを備え、

記憶部11は、ネイティブアプリケーションのテンプレートであるテンプレートアプリケーション111を記憶し、

ここで、テンプレートアプリケーション111は、スマートフォン等の携帯端末にインストールされると、所定のアクセス先にアクセスして情報を取得し、当該端末の表示部に表示させるネイティブアプリケーションのテンプレートであり、

テンプレートアプリケーション111は、設定情報112と、1以上のプログラムファイル113とを含んでいて、記憶部11の所定のフォルダ内に格納されており、

受付部12は、開発用端末2から、リクエスト用ページ30の取得要求を受け付けると、リクエスト用ページ30を開発用端末2に送信し、

ここで、リクエスト用ページ30には、入力欄として、ネイティブアプリケーションがアクセスするウェブアプリケーションのロケーションを示すアドレスの入力欄31が設けられ、また、ネイティブアプリケーションの表示態様情報に係る入力欄として、背景色の入力欄34、及びアイコン画像の入力欄35が設けられており、

入力欄31には、ウェブアプリケーションのメインページのURLが入力され、背景色の入力欄34には、カラーコードや背景画像を示すアドレスが入力され、アイコン画像の入力欄35には、画像ファイルのアドレスが入力され、

受付部12が、開発用端末2から、ウェブアプリケーションのメインページのアドレス、及び表示態様情報とともに、ネイティブアプリケーションの生成要求を示すリクエストを受け付けると、生成部13は、コピーして新たに生成されたテンプレートアプリケーション111に含まれる設定情報の内容を、受付部12が受け付けたウェブアプリケーションのメインページのアドレス、及び表示態様情報に基づいて書き換えてネイティブアプリケーションを生成し、

ここで、生成されたネイティブアプリケーションは、携帯端末にインストールされて実行されると、設定情報に含まれているウェブアプリケーションのアドレスに基づいて、ウェブアプリケーションに対応するウェブページを取得し、設定情報に含まれている表示態様情報に基づいて取得したウェブページの表示態様を変更して、携帯端末の表示部に表示させるものであり、

変換部14は、生成部13がネイティブアプリケーションを生成すると、当該ネイティブアプリケーションの設定情報及び1以上のプログラムファイルを格納している所定のフォルダを所定の方式で圧縮し、端末がインストール可能な形式のネイティブアプリケーションファイルに変換し、

送信部15は、ネイティブアプリケーションファイルを所定の記憶領域に格納し、当該ファイルのアドレスを表示する送信用ページを開発用端末2に送信し、開発用端末2において、送信用ページ40に設けられているダウンロードボタン41が押下されると、生成したネイティブアプリケーションファイルを開発用端末2に送信

する、

アプリケーション生成システム。」

イ 周知技術

(ア) 掌田津耶乃「Android Studioではじめる Androidプログラミング入門 第2版 Android 6.0 Marshmallow Android Studio 1.5対応」(株式会社秀和システム, 平成27年12月27日発行) (以下「引用文献2」という。甲2) の356頁～368頁には, 以下の記載がある。

a 「カメラを利用するためには, アプリケーションに「パーミッション (アクセス権)」を用意する必要があります。これは, AndroidManifest.xmlに記述します。AndroidManifest.xmlを開き, <manifest>タグ内の適当なところに以下のタグを追記してください……」 (356頁)

b 「7.3.3 GPS利用のパーミッション設定
……

GPSも, やはりパーミッション関係の情報を用意しなければ利用できないようになっています。AndroidManifest.xmlを開き, <manifest>タグ内に以下のタグを追加してください。」 (368頁)

(イ) 三宅 理「第7章 HTML5+JavaScriptでアプリ開発 はじめてのPhoneGap (その1) ここまでできるPhoneGap」Interface増刊 Smartphone World Volume. 4 (CQ出版株式会社, 平成24年6月1日発行) (以下「引用文献3」という。甲3, 乙1) の70頁～72頁には, 以下の記載がある。

「地図とGPSを組み合わせる

PhoneGapのすばらしさを体験するのに, 地図とGPSを組み合わせたアプリを例に紹介します。

・・・スマートフォン上に地図を表示します。さらに、PhoneGapの「Geolocation」を使用して、現在位置にマーカを表示します。」

(ウ) 「スマホはなぜ動くのか? 必ず悩むアプリ開発の6ポイントを徹底解説! Point1 ファイルがわかるとスマホアプリの仕組みが見えてくる」日経ソフトウェア2013年6月号(日経BP社,平成25年4月24日発行)(以下「引用文献4」という。甲4)の10頁~11頁,14頁,16頁には,以下の記載がある。

a 「設定ファイルは,そのアプリのアプリ名やバージョン,対応言語(英語や日本語etc.)などを記述しておくファイルです。」(10頁~11頁)

b 「設定ファイル「ios__sample-Info.plist」(5)のios__sample-Info.plistはアプリの設定ファイルです。アプリの名前やバージョン,そのアプリが利用するStoryboardのファイル等の情報が記述されています。」(14頁)

c 「超重要な設定ファイル「AndroidManifest.xml」(5)のAndroidManifest.xmlは,頻りに編集することになる超重要な設定ファイルです(図14)。この中ではアプリの名前やアイコン,対応するAndroidのバージョン,起動時に実行する○○Activityクラス,そのアプリをネットワークに接続できるようにするか?,SDカードにファイルを保存できるようにするか?などの,様々な項目を設定します。」(16頁)

(エ) 掌田津耶乃「HTML5とコピペでスマホアプリ開発者デビュー! Part2 いよいよデビュー スワイプ可能ミニゲームを作る」日経ソフトウェア2012年10月号(日経BP社,平成24年8月24日発行)(以下「引用文献5」という。甲5,乙2)の29頁には,以下の記載がある。

「横画面に固定する

スマートフォンでは端末が縦の姿勢のときと,横の姿勢のときで画面が切り替わります。ただ,ゲームでは画面が切り替わらない方が望ましいので,ここでは横画

面に固定する設定を施します。

Androidでは、AndroidManifest.xmlに「android:screenOrientation="landscape"」の1行を追加します(図5)。

iOSでは、左側のリストの青いアイコンのプロジェクト名(MatatabiPakkun)を選択すると、その右側に「TARGETS」という項目が現れるので、その下にある「MatatabiPakkun」を選びます。すると、プロジェクトの各種設定が表示されます。この中に「Supported Device Orientations」という項目があるので、「Landscape Left」か「Landscape Right」を選びます。

これで、画面を横向きに固定できます。なお、ここでいうorientationは「方向」、landscapeは「横方向」という意味です。」

(4) 「Androidエンジニア養成読本 現場で役立つノウハウと仕事にしたい人のための必須知識満載！」(株式会社技術評論社、平成23年11月25日発行)(以下「参考文献1」という。甲8)の157頁、162頁には、以下の記載がある。

a 「サンプルアプリケーションを起動すると、3Dグラフィクス界ではおなじみのティーポットの3Dモデルが表示されます。端末を傾けると視点が変化して、見る向きを変えることができます。」(157頁)

b 「■AndroidManifest.xml

加速度センサでデバイスの傾きを取得するため、傾けた際に画面のオリエンテーションが変わらないように、android:screenOrientation="portrait"を追加してportraitで固定します。」(162頁)

(2) 一致点及び相違点

本件補正発明と引用発明との一致点及び相違点は以下のとおりである。

ア 一致点

「携帯通信端末の所定の機能を実行させるためのパラメータに応じて、前記携帯通信端末において実行されるアプリケーションの動作を規定する設定ファイルを設定する設定部と、

前記設定ファイルに基づいてアプリケーションパッケージを生成する生成部と、を有するアプリケーション生成支援システム」である点。

イ 相違点

(ア) 相違点 1

設定ファイルを設定するパラメータが、

本件補正発明では、「携帯通信端末に固有のネイティブ機能を実行させるためのパラメータ」であるのに対して、

引用発明では、携帯通信端末の機能を実行させるためのパラメータではあるものの、携帯通信端末に固有のネイティブ機能を実行させるためのパラメータであることが特定されていない点。

(イ) 相違点 2

設定ファイルが、

本件補正発明では、「アプリケーションの、携帯通信端末の動きに伴う動作」を規定する「設定ファイル」であるのに対して、

引用発明では、「設定情報」が、ネイティブアプリケーションのウェブページ取得動作や表示動作を規定するものの、「携帯端末の動きに伴う」動作を規定するものであることが特定されていない点。

(3) 相違点についての判断

事案に鑑み、上記相違点 1 及び相違点 2 についてまとめて検討する。

ア(ア) 相違点 1 及び相違点 2 は、アプリケーションの特徴に係わる点に関連しており、本件補正発明のアプリケーション生成システムが生成するアプリケーションが、「ネイティブ機能を利用するアプリケーションであって、携帯通信端末の動きに伴う動作を行うアプリケーション」であるのに対して、引用発明のアプリケー

ション生成システムが生成するアプリケーションは、そのようなアプリケーションではなく、携帯端末にインストールされて実行されると、設定情報に含まれているウェブアプリケーションのアドレスに基づいて、ウェブアプリケーションに対応するウェブページを取得し、設定情報に含まれている表示態様情報に基づいて取得したウェブページの表示態様を変更して、携帯端末の表示部に表示させる点である。

(イ) ここで、ネイティブアプリケーションの設定ファイルに、引用発明に記載されている「ウェブアプリケーションのロケーション」、「ネイティブアプリケーションの表示態様情報」などのパラメータの他にも、以下の文献の記載にもあるように、ネイティブ機能のパラメータを含む各種のパラメータが設定されることが、本願出願前には当該技術分野における周知技術であった。

a 前記(1)イ(ア) a, bに記載されている「カメラを利用するためのパーミッション」、「GPS利用のパーミッション」などのネイティブ機能を実行させるためのパラメータ

b 前記(1)イ(ウ) a～cに記載されている「アプリの名前やバージョン、そのアプリが利用するStory boardのファイル等の情報」、「アプリの名前やアイコン、対応するAndroidのバージョン、起動時に実行する○○Activityクラス、そのアプリをネットワークに接続できるようにするか?、SDカードにファイルを保存できるようにするか?などの様々な項目」

c 前記(1)イ(エ)に記載されている「横画面に固定する設定」

d 前記(1)イ(オ) bに記載されている「加速度センサでデバイスの傾きを取得するため、傾けた際に画面のオリエンテーションが変わらないように固定するための設定」

(ウ) また、本件補正発明の「携帯通信端末の動きに伴う動作」について検討すると、本願の明細書（以下、図面を含めて「本件明細書」という。）の段落【0066】、【0076】、【0137】には、「携帯通信端末の動き」に関連すると考えられる構成として、「GPS」と「加速度センサ」が例示されている。

そして、本件補正発明の「携帯通信端末の動き」が「携帯通信端末の移動」であるとした場合、「携帯通信端末の移動に伴う動作」を行う「地図とGPSを組み合わせたアプリ」は、例えば、前記(1)イ(イ)のとおり、周知であり、GPSに関する前記(1)イ(ア)bの記載も参酌すると、上記「地図とGPSを組み合わせたアプリ」では、ネイティブ機能であるGPSに関連するパラメータを設定ファイルに設定しているものと認められる。

また、本件補正発明の「携帯通信端末の動き」が、端末を傾けるような「動き」であるとする、「携帯通信端末の傾き」に伴う動作を行うアプリは、例えば、前記(1)イ(オ)aに記載されているように周知であり、前記(1)イ(オ)bの記載を参照すると、このアプリでは、ネイティブ機能である加速度センサに関連するパラメータが設定ファイルに設定されると認められる。

そうすると、「携帯通信端末の動きに伴う動作を行うアプリ」は、本願の出願前に当該技術分野において周知であり、加えて、当該アプリが利用するネイティブ機能のパラメータを設定ファイルに設定可能とすることも、本願の出願前には当該技術分野における周知技術であった。

(エ) 引用発明は、「リクエスト用ページ30には、入力欄として、ネイティブアプリケーションがアクセスするウェブアプリケーションのロケーションを示すアドレスの入力欄31が設けられ、また、ネイティブアプリケーションの表示態様情報に係る入力欄として、背景色の入力欄34、及びアイコン画像の入力欄35が設けられており」、「入力欄31には、ウェブアプリケーションのメインページのURLが入力され、背景色の入力欄34には、カラーコードや背景画像を示すアドレスが入力され、アイコン画像の入力欄35には、画像ファイルのアドレスが入力される」ものであるところ、引用文献1の段落【0010】の「前記テンプレートアプリケーションは、前記アクセス先を設定する設定情報を含み、前記生成部は、前記受付部が前記リクエストを受け付けたことに応じて、前記テンプレートアプリケーションをコピーし、コピーされた前記テンプレートアプリケーションに含まれる

前記設定情報の前記アクセス先を前記受付部が受け付けた前記アドレスに設定することで前記ネイティブアプリケーションを生成する」との記載及び「この発明によれば、アプリケーション生成装置は、設定情報に含まれているアクセス先を、受付部が受け付けたアドレスに変更するという単純な処理によってネイティブアプリケーションを容易に生成することができる」との記載も考慮すると、引用発明は、ネイティブアプリケーションの設定情報（設定ファイル）のパラメータを、リクエスト用ページ30の入力欄、すなわち、GUIを用いて簡単に設定することで、ネイティブアプリケーションを容易に生成できるアプリケーション生成システムであるといえる。

そして、アプリケーション生成システムを用いて「携帯端末」の「ネイティブアプリケーション」の開発を行う当業者であれば、「携帯通信端末の動きに伴う動作を行うアプリ」を利用可能であるところ、この周知の「携帯通信端末の動きに伴う動作を行うアプリ」を生成する場合でも、設定ファイルにネイティブ機能のパラメータをGUIを用いて簡単に設定することで、アプリを容易に生成するという課題を解決できることは当然に予測し得たものであるから、引用発明のアプリケーション生成システムを、上記周知の「携帯通信端末の動きに伴う動作を行うアプリ」の生成に用いる動機はあったといえる。

(オ)したがって、引用発明のアプリケーション生成システムを、「携帯通信端末の動き（すなわち、携帯通信端末の移動や傾きなどの動き）に伴う動作」を行うアプリケーションの生成に用い、その際、当該アプリケーションの設定ファイルに、当該アプリケーションが利用するネイティブ機能のパラメータを設定するように構成すること、すなわち、引用発明において、アプリケーションの設定ファイルを、「アプリケーションの、携帯通信端末の動きに伴う動作を規定する設定ファイル」とし、当該設定ファイルを設定するためのパラメータを、GPSや加速度センサのような携帯通信端末の動きに伴う動作に関連したネイティブ機能を実行させるためのパラメータとすることは、当業者が容易に想到し得たことである。

イ 本件補正発明の奏する作用効果は、引用発明と引用文献2～5及び参考文献1に記載された周知技術の奏する作用効果から予測される範囲内のものにすぎず、格別顕著なものということとはできない。

ウ よって、本件補正発明は、引用発明と引用文献2～5及び参考文献1に記載された周知技術に基づいて、当業者が容易に発明をすることができたものであり、特許法29条2項により、特許出願の際独立して特許を受けることができないものであるので、本件補正は却下すべきものである。

(4) 本件補正前発明について

本件補正前発明は、本件補正発明から、「通信端末」に係る限定事項と、「前記通信端末において実行されるアプリケーションの動作」に係る限定事項を削除したものである。

本件補正前発明の発明特定事項をすべて含み、さらに他の事項を付加したものに相当する本件補正発明が、引用発明と引用文献2～5及び参考文献1に記載された周知技術に基づいて、当業者が容易に発明をすることができたのであるから、本件補正前発明も、引用発明と引用文献2～5及び参考文献1に記載された周知技術に基づいて、当業者が容易に発明をすることができたものである。

したがって、本件補正前発明は、特許法29条2項により特許を受けることができない。

第3 原告主張の取消事由

1 取消事由1（本件補正発明についての容易想到性の判断の誤り）

(1) 引用発明のアプリケーション生成システムを、「携帯通信端末の動きに伴う動作を行うアプリ」の生成に用いる動機はないこと

ア 引用文献1の段落【0002】、【0004】～【0007】、【0023】、【0024】によると、引用発明のアプリケーション生成システムは、アプリケーションサーバにおいてネイティブアプリケーションを検索したユーザに、CMSにより開発したウェブアプリケーションを利用してもらえないという問題

に鑑み（段落【0002】、【0004】、【0005】）、CMSによって構築したウェブアプリケーションと同等の機能を有するネイティブアプリケーションを開発することを提案している（段落【0006】）。引用発明は、その提案を前提とした上で、ネイティブアプリケーションの開発には多大な開発工数が必要であることから、ネイティブアプリケーションを容易に生成することができるアプリケーション生成装置を提供することを課題としている（段落【0006】、【0007】）。

ここで、ウェブアプリケーションは、一般的に、Webサーバ上で動作し、Webブラウザを用いて利用するアプリケーションとして認識されている（甲22）。すなわち、ウェブアプリケーションは、入力されたデータの管理や、データに基づいて出力する情報を組み立てる処理をインターネット上のWebサーバが行い、ユーザによる情報の入力や、出力された情報の画面表示といった処理を端末側のWebブラウザが行うアプリケーションである（甲23）。

このように、ウェブアプリケーションを利用するに当たり、ユーザが操作する端末側の担う処理は、入力されたデータの送信処理と受信したデータの表示処理のみであり、データ処理等の実体的な部分の演算処理は、もっぱらWebサーバ側に任せて実行する構造となっている。

また、ネイティブアプリケーションは、一般的に、端末内の演算装置が直接に演算処理を行う（実行する）タイプのアプリケーションとして認識されている（甲24）。引用文献1の段落【0004】、【0023】に記載されているとおり、ネイティブアプリケーションは、端末内で実行されるという性質上、アプリケーションサーバ等のアプリケーション提供元からダウンロードして端末にインストールされる。

以上のとおり、ウェブアプリケーションとネイティブアプリケーションとは、技術的に全く異なるタイプのアプリケーションであり、引用発明は、アプリケーションサーバにおいてネイティブアプリケーションを検索したユーザに、CMSにより開発したウェブアプリケーションを利用してもらうことを前提とした上で、多大な開発工数を要することなく、ウェブアプリケーションと同等の機能を有するネイテ

ィブアプリケーションを容易に生成するという課題を解決するものである。

イ 引用文献1の段落【0025】、【0026】、【0029】～【0031】、【0034】～【0036】によると、引用発明のアプリケーション生成システムで生成されたネイティブアプリケーションは、「設定情報」に設定された「ウェブアプリケーションのロケーションを示すアドレス」、「メニューページに表示するメニュー画像のファイルの格納先アドレス」、「メニュー画像に対応するリンク先のページのアドレス」、「カラーコード（RGB値を十六進法で表記した文字列）や背景画像を示すアドレス」等のパラメータに応じて、指定された表示態様で指定されたウェブアプリケーションを表示するものであり、上記パラメータは、いずれもスマートフォン等の携帯通信端末の画面にウェブページを表示するためのパラメータに限られており、表示対象のロケーション（所在）を示すためのアドレスのみである。

このように、引用発明のアプリケーション生成システムは、ウェブアプリケーションの利用、すなわち、ブログ等の既存のウェブアプリケーションにアクセスし、その表示態様を指定して表示する機能を備えたネイティブアプリケーションを生成するものであるから、表示したいウェブアプリケーションのロケーションを示すアドレスや、所望の背景画像のロケーションを示すアドレス等の簡単なパラメータをGUIに入力するだけで、所望のウェブアプリケーションを表示するネイティブアプリケーションを生成できるのである。

以上のように、引用発明のアプリケーション生成システムは、既存のウェブアプリケーションを利用して所望のコンテンツ（ブログ等）を表示するという機能に特化した構成を備えている。

ウ 引用発明のアプリケーション生成システムで生成されたネイティブアプリケーションは、ブログ等の既存のウェブアプリケーションにアクセスできるだけであり、Webサーバ上のブログ等のコンテンツを書き換えるものではない。したがって、仮に、設定情報にネイティブ機能（例えば、GPS、加速度センサ等）を実行するためのパラメータをGUIを用いて簡単に設定できたとしても、携帯通信

端末で動作するのは表示機能に限定されたWebブラウザにすぎないから、「携帯通信端末に固有のネイティブ機能に応じて携帯通信端末の動きに伴う動作を行うネイティブアプリケーション」を容易に生成できるとはいえない。

したがって、当業者は、設定ファイルにネイティブ機能のパラメータをGUIを用いて簡単に設定することで、アプリを容易に生成するという課題を解決できることは当然に予測し得たということとはできない。

エ(ア) 前記ア、イのとおり、引用発明のアプリケーション生成システムは、既存のウェブアプリケーションを利用しつつ所望の表示態様でコンテンツを表示するという機能に特化した構成を備えているから、引用発明のアプリケーション生成システムを用いてネイティブアプリケーションを生成するためには、目的とするコンテンツを有する既存のウェブアプリケーションが特定されなければならない。引用発明に接した当業者が、ブログ、ショッピングサイト等のコンテンツを表示する既存のウェブアプリケーションを利用するという目的を離れて、引用発明のアプリケーション生成システムを用いて「携帯通信端末の動きに伴う動作を行うアプリ」を生成することはない。

また、前記ウのとおり、当業者は、引用発明のアプリケーション生成システムを用いることにより「設定ファイルにネイティブ機能のパラメータをGUIを用いて簡単に設定することで、アプリを容易に生成するという課題を解決できる」とは考えないから、引用発明のアプリケーション生成システムを用いて「携帯通信端末の動きに伴う動作を行うアプリ」を生成することはない。

(イ) 仮に、当業者が引用発明のアプリケーション生成システムを用いて「携帯通信端末の動きに伴う動作を行うアプリ」を生成しようと試みたとしても、その際には、既存の「携帯通信端末の動きに伴う動作を行うウェブアプリケーション」のロケーションを示すアドレスを取得する必要があるところ、「携帯通信端末の動きに伴う動作を行う」ためには、携帯通信端末の動きを検知して、その動きに応じて動作する機能を備えている必要があるが、ウェブアプリケーションを利用する際

に端末側で実行されるWebブラウザには、携帯通信端末の動きを検出する機能はないから、既存のウェブアプリケーションでは「携帯通信端末の動きに伴う動作を行う」ことはできない。

(ウ)したがって、当業者が引用発明のアプリケーション生成システムを用いて「携帯通信端末の動きに伴う動作を行うアプリ」を生成することはない。

オ 引用発明の課題は、「アプリケーションサーバにおいてネイティブアプリケーションを検索したユーザに、CMSにより開発したウェブアプリケーションを利用してもらうこと」(段落【0005】)と「ネイティブアプリケーションを容易に生成することができるアプリケーション生成装置を提供すること」である(段落【0006】、【0007】)。

本件審決は、ネイティブアプリケーションの設定ファイルにネイティブ機能のパラメータを含む各種のパラメータが設定されることは周知技術(以下「周知技術1」という。)であるとし、また、携帯通信端末の動きに伴う動作を行うアプリ及び当該アプリが利用するネイティブ機能のパラメータを設定ファイルに設定することは周知技術(以下「周知技術2」という。)であるとしている。

周知技術1は、ネイティブアプリケーションに、ネイティブ機能の実行を含む各種の設定を行う場合には、設定ファイルにネイティブ機能のパラメータを含む各種のパラメータを設定するという技術である。すなわち、周知技術1の課題は、「ネイティブアプリケーションにネイティブ機能の実行を含む各種の設定を行うこと」である。

また、周知技術2は、設定ファイルにネイティブ機能のパラメータを設定することにより携帯通信端末の動きに伴う動作を行うアプリケーションである。すなわち、周知技術2の課題は、「携帯通信端末の動きに伴う動作を行うアプリケーションを提供すること」である。

このように、引用発明が解決しようとする課題と周知技術1及び周知技術2が解決しようとする課題との間に共通性はない。

カ 前記ア～エのとおり、ネイティブアプリケーションの開発を行う当業者は、引用発明のアプリケーション生成システムを、「携帯通信端末の動きに伴う動作を行うアプリ」の生成に用いる動機はない。

また、前記オのとおり、引用発明が解決しようとする課題と周知技術（周知技術1及び周知技術2）が解決しようとする課題との間に共通性はないため、その観点からも、引用発明のアプリケーション生成システムを、「携帯通信端末の動きに伴う動作を行うアプリ」の生成に用いる動機はない。

したがって、引用発明のアプリケーション生成システムを、上記周知の「携帯通信端末の動きに伴う動作を行うアプリ」の生成に用いる動機はあるとした本件審決の判断には誤りがある。

(2) 引用発明において、アプリケーションの設定ファイルを、「アプリケーションの、携帯通信端末の動きに伴う動作を規定する設定ファイル」とし、当該設定ファイルを設定するためのパラメータを、GPSや加速度センサのような携帯通信端末の動きに伴う動作に関連したネイティブ機能を実行させるためのパラメータとすることは、当業者が容易に想到し得たことではないこと

ア 本件審決は、引用発明のアプリケーション生成システムを、携帯通信端末の動き（携帯通信端末の移動や傾きなどの動き）に伴う動作を行うアプリケーションの生成に用いることを前提にして、引用発明と周知技術1及び周知技術2を組み合わせたが、前記(1)のとおり、携帯通信端末用のネイティブアプリケーションの開発を行う当業者は、引用発明のアプリケーション生成システムを用いて、「携帯通信端末の動きに伴う動作を行うアプリ」の生成を試みる動機はない。

イ 引用発明のアプリケーション生成システムにおいて、「当該設定ファイルを設定するためのパラメータ」、すなわちGUIにて設定するパラメータを「携帯通信端末に固有のネイティブ機能を実行させるためのパラメータ」に置き換えてしまうと、利用すべきウェブアプリケーションのロケーションを特定することができなくなり、引用発明が前提とする「ウェブアプリケーションを利用するネイティブ

アプリケーション」を生成することができなくなってしまう。

ウ このように、引用発明と上記各周知技術とを組み合わせる動機付けがなく、また、引用発明と上記各周知技術の組合せに阻害要因があるが、本件審決の判断は、この点を看過しており、誤りである。

(3) 本件補正発明の奏する作用効果は、引用発明と引用文献2～5及び参考文献1に記載された周知技術の奏する作用効果から予測される範囲のものではなく、格別顕著なものであること

本件補正発明の作用効果は、携帯通信端末に固有のネイティブ機能に応じて携帯通信端末の動きに伴う動作を行うネイティブアプリケーションを、少ない開発負担で生成することができるアプリケーション生成支援システムを提供できることである。

前記(1)及び(2)のとおり、引用発明と引用文献2～5及び参考文献1に記載された周知技術とを組み合わせる動機がない以上、本件補正発明が奏する作用効果が予測されるはずもない。仮に、それらを組み合わせることができたとしても、引用発明が既存のウェブアプリケーションの利用を前提としている限り、既存のウェブアプリケーションでは実現し得ない動作、すなわち「携帯通信端末の動きに伴う動作」を実現するネイティブアプリケーションを生成することまでは想定できない。

したがって、本件補正発明の奏する作用効果は、引用発明と引用文献2～5及び参考文献1に記載された周知技術の奏する作用効果から予測される範囲のものではなく、格別顕著なものであるから、これと異なる本件審決の判断は、誤っている。

(4) 被告の主張について

ア 被告は、引用文献1の段落【0005】、【0006】の記載に基づき、引用発明の課題は、CMS（コンテンツ管理システム）によって構築したウェブアプリケーションと同等の機能を有するネイティブアプリケーションを容易に開発して、アプリケーションサーバにおいてネイティブアプリケーションとして検索可能とすることであると主張する。

しかし、段落【0006】においては、「CMSにより開発したウェブアプリケーションを利用してもらうことができないという問題」の解決手段として、CMSによって構築したウェブアプリケーションと同等の機能を有するネイティブアプリケーションを新たに開発することは、適切ではないことを指摘しているのである。

引用文献1の段落【0007】、【0025】の記載からすると、引用発明は、アプリケーションサーバにおいてネイティブアプリケーションを検索したユーザに、CMSにより開発したウェブアプリケーション（既存のウェブアプリケーション）を利用してもらうことを前提とし、そのためのネイティブアプリケーションを（開発という手段を採らずに）容易に生成することを課題としてされたものである。

イ 被告は、PhoneGapは、HTMLやJavaScriptなどのWeb系技術を用いてウェブアプリケーションを作る感覚で容易に作成可能であって、App Store（アプリケーションサーバ）等で配布可能となるアプリを作成可能なフレームワークであるから、引用発明の課題を解決する際に利用するには好適なフレームワークであると主張する。

しかし、PhoneGapは、あくまでWeb系技術を用いてネイティブアプリケーションを新たに開発するためのフレームワークであり、目的とするネイティブアプリケーションを新たに開発するためには、HTMLやJavaScript等を用いてソースコード（プログラム）を書く必要があり、PhoneGapを用いることは、目的の機能を備えたネイティブアプリケーションを新たに開発する（新たにソースコードを書く）ことに他ならない。

これに対し、引用発明は、既存のウェブアプリケーションを利用するためのネイティブアプリケーションを容易に生成することを課題とするものであり、PhoneGapのようにソースコードを書くという面倒な作業を要することなく、利用したいウェブアプリケーションのアクセス先アドレス等の情報を入力するだけで、ネイティブアプリケーションを容易に生成することができるのである。

したがって、既存のアプリケーションを利用するのではなく、同等の機能を有す

るネイティブアプリケーションを新たに開発するための手段である P h o n e G a p は、引用発明の課題を解決するものではないから、引用発明の課題を解決する際に利用するに当たって好適なフレームワークではない。

ウ 被告は、当業者が引用発明を実装するに当たって、アプリ作成フレームワークである P h o n e G a p を用いる動機付けがあると主張する。

しかし、被告は、引用発明の前提（既存のウェブアプリケーションを利用してもらうこと）を看過し、引用発明の課題を誤って認定している。そして、誤って認定した課題に基づいて、ウェブアプリケーションと同等の機能を有するネイティブアプリケーションを新たに開発するために、P h o n e G a p が好適であるという誤った結論に至っている。

引用発明は、既存のアプリケーションを利用するためのネイティブアプリケーションを容易に生成する技術であるところ、P h o n e G a p は、W e b 系技術を用いて新たにソースコードを書くことにより、既存のアプリケーションと同等の機能を有するネイティブアプリケーションを新たに開発するためのフレームワークであるから、P h o n e G a p を用いた場合、既存のウェブアプリケーションを利用することはできない。

そして、既存のウェブアプリケーションを利用することができない技術を、既存のアプリケーションを利用する技術に適用することはできないから、引用発明に対して P h o n e G a p を適用するには、阻害要因がある。

また、引用発明に対して P h o n e G a p を適用したとすると、目的とするネイティブアプリケーションのソースコードを新たに書くという作業が発生するから、引用発明の課題である既存のウェブアプリケーションを利用するためのネイティブアプリケーションを容易に生成することが達成できなくなってしまう。この観点からも、当業者が引用発明を実装するに当たって、アプリ作成フレームワークである P h o n e G a p を用いる動機付けはない。

エ 被告は、スマートフォンでは端末が縦の姿勢のときと、横の姿勢のとき

で画面が切り替わることは「携帯通信端末の動きに伴う動作」に他ならず、「横画面に固定する設定」や「縦画面に固定する設定」は、いずれも、「アプリケーションの、携帯通信端末の動きに伴う動作」を規定するものであるといえ、Androidにおける、AndroidManifest.xmlファイルは「アプリケーションの、携帯通信端末の動きに伴う動作」を規定する設定ファイルに相当するものであるといえと主張する。

しかし、PhoneGapの「横画面に固定する設定」や「縦画面に固定する設定」は、あくまで携帯通信端末の動きにかかわらず、単に横画面又は縦画面に一律に画面の方向を固定するものであるから、「携帯通信端末の動きに伴う動作」を規定するものではない。

オ 被告は、引用発明のウェブアプリケーションは、HTMLやJavaScriptで記述されているところ、JavaScriptは、Webブラウザで実行される制御コードを記述するためのスクリプト言語であるから、コンテンツを既存のウェブサイトから取得しているとしても、当該コンテンツに対する制御処理は端末側で実行されるWebブラウザにおいて、JavaScriptに従って行われているといえ、したがって、引用発明は、既存のウェブアプリケーションを利用して所望のコンテンツ（ブログ等）を表示するという機能に特化したものとはいえないと主張する。

しかし、本件審決において認定された引用発明は、ウェブアプリケーションがHTMLやJavaScriptで記述されていること、及び、既存のウェブサイトから取得したコンテンツに対する制御処理が端末側で実行されるWebブラウザにおいてJavaScriptに従って行われていることを構成として有していないから、被告の上記主張は、本件審決において認定された引用発明に基づかないものであり、失当である。

本件審決において認定された引用発明は、「ここで、生成されたネイティブアプリケーションは、携帯端末にインストールされて実行されると、設定情報に含まれて

いるウェブアプリケーションのアドレスに基づいて、ウェブアプリケーションに対応するウェブページを取得し、設定情報に含まれている表示態様情報に基づいて取得したウェブページの表示態様を変更して、携帯端末の表示部に表示させるものであり」という構成を有しており、ウェブアプリケーションに対応するウェブページを取得して表示することが特定されているのみである。

カ 被告は、PhoneGapは、引用発明と同様に、HTMLやJavaScript等で記述されたウェブアプリケーションからスマートフォン用アプリを生成可能であると主張する。

しかし、PhoneGapは、HTMLやJavaScriptといったWeb系技術を用いて目的とするネイティブアプリケーションのソースコード（プログラム）を、開発者が自ら新規に書かなければならない。PhoneGapは、既存のウェブアプリケーションからネイティブアプリケーションを容易に生成する技術ではなく、一般的なWeb系技術を用いて、ネイティブアプリケーションをプログラミングすることが可能な技術にすぎない。

キ 被告は、PhoneGapは、設定ファイル（AndroidManifest.xmlなど）にネイティブ機能（端末の加速度センサを用いた画面の回転制御）の制御内容を規定するためのパラメータ（landscapeやportrait）を設定していると主張する。

しかし、前記エのとおり、PhoneGapにおける「横画面に固定する設定（landscape）」や「縦画面に固定する設定（portrait）」は、携帯通信端末の動きとは無関係に設定されるものであり、「携帯通信端末の動きに伴う動作」を規定するものではない。

ク 被告は、引用発明のアプリケーション生成システムにPhoneGapを適用する場合には、「当該設定ファイルを設定するためのパラメータ」を「携帯通信端末に固有のネイティブ機能を実行させるためのパラメータ」に置き換えてしまうのではなく、ウェブサーバで提供されるコンテンツにアクセスする際に使用され

る「当該設定ファイルを設定するためのパラメータ」とは別の「携帯通信端末に固有のネイティブ機能を実行させるためのパラメータ」を追加設定するだけであるから、引用発明が前提とする「ウェブアプリケーションを利用するネイティブアプリケーション」を生成することができなくなることはないと主張する。

しかし、被告は、「携帯通信端末に固有のネイティブ機能を実行させるためのパラメータ」を具体的にどのように追加するのか立証していないから、被告の上記主張は根拠がない。

ケ 被告は、一般的なウェブサイト表示用のウェブアプリケーションは、加速度センサを備えたスマートフォン等の端末では、加速度センサによって検出した端末の姿勢に基づいて、端末の姿勢に基づいた画面回転表示制御を行っているから、一般的なウェブサイト表示用のウェブアプリケーションにおいても、「携帯通信端末の動きに伴う動作」(端末の姿勢に基づいた画面回転表示制御)を実現することが可能であると主張する。

しかし、端末の姿勢は、あくまで端末に備えられた加速度センサで検出するものであるから、サーバ側で動作するウェブアプリケーションにおいては、端末側の姿勢を知る由もない。端末の姿勢に基づいた画面回転表示制御は、ウェブアプリケーションの機能ではなく、端末側の機能にすぎない。

したがって、一般的なウェブサイト表示用のウェブアプリケーションにおいて、「携帯通信端末の動きに伴う動作」(端末の姿勢に基づいた画面回転表示制御)を実現することは可能であるとする被告の上記主張は、誤りである。

2 取消事由2 (本件補正前発明についての容易想到性の判断の誤り)

本件補正前発明についても、引用発明において、アプリケーションの設定ファイルを、「アプリケーションの動作を規定する設定ファイル」に変更し、当該設定ファイルを設定するためのパラメータを、GPSや加速度センサのような携帯通信端末に固有のネイティブ機能を実行させるためのパラメータに変更することには、阻害要因があり、引用発明と引用文献2～5及び参考文献1に記載された周知技術とを

組み合わせることはできない。

したがって、本件補正前発明は、当業者が引用発明と引用文献2～5及び参考文献1に記載された周知技術に基づいて容易に想到し得たものではない。

第4 被告の主張

1 取消事由1について

(1) 周知技術であるPhone Gapの概要

引用文献3（66頁表題，66頁副題，66頁中欄下から2行～右欄3行，66頁表1，66頁右欄13行～67頁左欄13行，70頁右欄6行～最終行，72頁左欄1行～25行），引用文献5（16頁左欄7行～27行，29頁左欄1行～16行，36頁左下欄下から10行～末行），三宅理「スマートフォン開発倶楽部第5回」WEB+DB PRESS Vol. 66（株式会社技術評論社，平成24年1月25日発行）（以下「乙3文献」という。甲7，乙3）（55頁左欄10行～14行，159頁左欄1行～右欄4行，160頁右欄1行～7行）の記載からすると，周知技術のPhone Gapは以下のような技術であるといえる。

「iPhoneやAndroid，Windows Phoneなどのマルチプラットフォームに対応するアプリを作成可能なフレームワークであって，

HTMLやJavaScriptなどのWeb系技術を用いてWebアプリを作る感覚で作成可能であり，

JavaScriptでPhone GapのAPI（ライブラリの関数）を呼び出せば，プラグインの仕組みを使って，カメラやGPSなど端末のネイティブ部分にアクセス可能であり，

端末のGPS機能にアクセスすることで，GPSで取得した端末の現在位置が中心となるように地図を表示することが可能となり，

スマートフォンでは加速度センサを用いて端末の姿勢を検知し，端末が縦の姿勢のときと，横の姿勢のときで画面が切り替わるところ，横画面に固定する設定を施す場合，Androidでは，AndroidManifest.xmlに「an

android:screenOrientation =” landscape”」の1行を追加することで、iOSでは、「Landscape Right」または「Landscape Left」のアイコンを選択することで横画面に固定可能となり、縦画面に固定する設定を施す場合、Androidでは、AndroidManifest.xmlに「android:screenOrientation =” portrait”」の1行を追加することで、iOSでは、「portrait」のアイコンを選択することで、縦画面に固定可能となり、

作成されたアプリの実体はブラウザUIで動くアプリであって、App Store等で配布可能となるアプリを作成可能なフレームワークであるPhoneGap」

(2) 引用発明への周知技術のPhoneGapの適用

ア 動機付け

引用文献1の段落【0004】～【0007】によると、引用発明の課題は、CMS（コンテンツ管理システム）によって構築したウェブアプリケーションと同等の機能を有するネイティブアプリケーションを容易に開発して、アプリケーションサーバにおいてネイティブアプリケーションとして検索可能とすることであるといえる。

ここで、PhoneGapは、HTMLやJavaScriptなどのWeb系技術を用いてWebアプリ（ウェブアプリケーション）を作る感覚で容易に作成可能であって、App Store（アプリケーションサーバ）等で配布可能となるアプリ（App Store等で配布可能となれば、アプリケーションサーバ（App Store等）においてネイティブアプリケーションとして検索可能となる）を作成可能なフレームワークであるから、引用発明の課題を解決する際に利用するには好適なフレームワークであるといえる。

したがって、当業者が引用発明を実装するに当たって、アプリ作成フレームワークであるPhoneGapを用いる動機付けがあるといえる。

イ(ア) PhoneGapでは、JavaScriptでPhoneGapのAPI（ライブラリの関数）を呼び出せば、プラグインの仕組みを使って、GPSなど端末のネイティブ部分にアクセス可能であり、端末のGPS機能にアクセスすることで、GPSで取得した端末の現在位置が中心となるように地図を表示することが可能となるから、引用発明において地図表示アプリケーションを生成する際に、PhoneGapのフレームワークを採用することで、ネイティブ機能であるGPS機能が利用可能となり、携帯通信端末の動き（移動）に伴う動作（GPSで取得した端末の現在位置が中心となるように地図を表示する地図表示制御）を設定可能となる。

(イ) また、スマートフォンでは、ネイティブ機能である加速度センサから取得した値を用いて端末姿勢を検出しており、端末が縦の姿勢のときと、横の姿勢のときで画面が切り替わるころ、PhoneGapでは、横画面に固定する設定を施す場合、Androidでは、AndroidManifest.xmlに「`android:screenOrientation = "landscape"`」の1行を追加することで、iOSでは、「`Landscape Right`」又は「`Landscape Left`」のアイコンを選択することで横画面に固定可能となり、縦画面に固定する設定を施す場合、Androidでは、AndroidManifest.xmlに「`android:screenOrientation = "portrait"`」の1行を追加することで、iOSでは、「`portrait`」のアイコンを選択することで、縦画面に固定可能となる。

ここで、「スマートフォンでは端末が縦の姿勢のときと、横の姿勢のときで画面が切り替わる」ことは「携帯通信端末の動きに伴う動作」に他ならず、「横画面に固定する設定」や「縦画面に固定する設定」は、いずれも、「アプリケーションの、携帯通信端末の動きに伴う動作」を規定するものであるといえ、Androidにおける、AndroidManifest.xmlファイルは「アプリケーションの、携帯通信端末の動きに伴う動作」を規定する「設定ファイル」に相当するものであ

るといえる。

ウ したがって、引用発明において、アプリケーションを生成する際に、P h o n e G a p のフレームワークを利用することで、本件補正発明と同様に、「携帯通信端末に固有のネイティブ機能を実行させるためのパラメータ」に応じて、携帯通信端末において実行される「アプリケーションの、携帯通信端末の動きに伴う動作」を規定する「設定ファイル」を備えることとなる。

(3) 本件補正発明の作用効果について

地図表示のウェブアプリケーションは、GPS機能を備えた端末では、GPS機能を用いて現在位置を取得し、現在位置を中心とした地図表示を行うことが可能であるから、地図表示のウェブアプリケーションにおいても、「携帯通信端末の動きに伴う動作」(端末の移動に伴って変化する現在位置を中心とした地図表示制御)を実現することが可能である(乙4の130頁～134頁)。

また、一般的なウェブサイト表示用のウェブアプリケーションは、加速度センサを備えたスマートフォン等の端末では、加速度センサによって検出した端末の姿勢に基づいて、端末の姿勢に基づいた画面回転表示制御を行っているから、一般的なウェブサイト表示用のウェブアプリケーションにおいても、「携帯通信端末の動きに伴う動作」(端末の姿勢に基づいた画面回転表示制御)を実現することが可能である。

したがって、引用発明にP h o n e G a p を組み合わせることで、これらのウェブアプリケーションから「携帯通信端末の動きに伴う動作」を実現するネイティブアプリケーションを生成することが想定可能であるといえるから、本件補正発明の作用効果は格別なものではない。

(4) 原告の主張について

ア 原告は、引用発明は、既存のウェブアプリケーションを利用して所望のコンテンツ(ブログ等)を表示するという機能に特化したものであると主張する。

しかし、引用文献1の段落【0024】によると、引用発明のウェブアプリケー

ションが、HTMLやJavaScriptで記述されており、引用発明のウェブアプリケーションの例として、ブログだけでなく、ゲームサイト等も示されている。

ここで、JavaScriptは、Webブラウザで実行される制御コードを記述するためのスクリプト言語である（乙4の26頁下から4行～最終行）から、コンテンツを既存のウェブサイトから取得しているとしても、当該コンテンツに対する制御処理は端末側で実行されるWebブラウザにおいて、JavaScriptに従って行われているといえる。ウェブアプリケーションのゲームでは、一般的に、ユーザからの操作入力を受け付けて、JavaScriptにより記述された制御コードを端末側で実行することでインタラクティブな制御処理を可能としている。

したがって、引用発明は、既存の「ウェブアプリケーション」を利用して所望のコンテンツ（ブログ等）を表示するという機能に特化したものとはいえないから、原告の上記主張は誤りである。

イ 原告は、設定ファイルにネイティブ機能のパラメータをGUIを用いて簡単に設定することで、アプリを容易に生成するという課題を解決できることは当然に予測し得たものでないと主張する。

しかし、前記(1)のとおり、PhoneGapは、引用発明と同様に、HTMLやJavaScript等で記述されたウェブアプリケーションからスマートフォン用アプリを生成可能であり、端末のブラウザUI（Webブラウザ）に、スマートフォンのネイティブ機能（GPSやカメラなど）にアクセスするためのJavaScript関数を実行させることで、プラグインの仕組みを使って、スマートフォンのネイティブ機能（GPSやカメラなど）にアクセス可能となる。また、設定ファイル（AndroidManifest.xmlなど）にネイティブ機能（端末の加速度センサを用いた画面の回転制御）の制御内容を規定するためのパラメータ（パラメータlandscapeやportrait）を設定している。

したがって、引用発明にPhoneGapを適用することにより、設定ファイルにネイティブ機能のパラメータをGUIを用いて簡単に設定することで、アプリを

容易に生成するという課題を解決できることは当然に予測し得たものであるといえるから、原告の上記主張は誤りである。

ウ 原告は、引用発明のアプリケーション生成システムにおいて、「当該設定ファイルを設定するためのパラメータ」すなわちG U Iにて設定するパラメータを、「携帯通信端末に固有のネイティブ機能を実行させるためのパラメータ」に置き換えてしまうと、利用すべきウェブアプリケーションのロケーションを特定することができなくなり、引用発明が前提とする「ウェブアプリケーションを利用するネイティブアプリケーション」を生成することができなくなってしまうから、引用発明と上記各周知技術との組合せに阻害要因がある旨主張する。

しかし、引用発明のアプリケーション生成システムにP h o n e G a pを適用する場合には、「当該設定ファイルを設定するためのパラメータ」を「携帯通信端末に固有のネイティブ機能を実行させるためのパラメータ」に置き換えてしまうのではなく、ウェブサーバで提供されるコンテンツにアクセスする際に使用される「当該設定ファイルを設定するためのパラメータ」とは別の「携帯通信端末に固有のネイティブ機能を実行させるためのパラメータ」を追加設定するだけであるから、引用発明が前提とする「ウェブアプリケーションを利用するネイティブアプリケーション」を生成することができなくなることはない。

したがって、原告の上記主張は誤りである。

2 取消事由2について

前記1(4)ウのとおり、引用発明のアプリケーション生成システムとP h o n e G a pとの組合せに阻害要因はないから、本件補正前の発明は、当業者が引用発明と引用文献2～5及び参考文献1等に記載された周知技術に基づいて容易に想到し得たものである。

したがって、本件審決における、本件補正前発明についての容易想到性の判断に誤りはない。

第5 当裁判所の判断

1 本件明細書の記載

本件明細書には、以下の記載がある（甲9の1・3）。

【技術分野】【0001】本発明は、アプリケーション生成支援システムおよびアプリケーション生成支援プログラムに関する。

【背景技術】【0002】モバイル用通信端末のアプリケーションとして、さまざまなアプリケーションが開発されている。モバイル用通信端末のアプリケーションは、Webアプリケーション、ネイティブアプリケーション、およびハイブリッドアプリケーションに分類される。Webアプリケーションは、演算処理をネットワーク上のサーバで実行するタイプのアプリケーションである。ネイティブアプリケーションは、演算処理を通信端末上でのみ実行するタイプのアプリケーションである。ネイティブアプリケーションは、例えば、カメラ、GPS、マイク、加速度センサなど、通信端末に固有のネイティブ機能を利用することができる、という特徴を有するアプリケーションである。ハイブリッドアプリケーションは、ネイティブ機能を利用ことができ、マルチプラットフォームに対応したアプリケーションを作成することができ、アプリケーションをwebブラウザのように使用するWeb Viewを用いることができるという特徴を有するアプリケーションである。

【0003】上記のように、ハイブリッドアプリケーションは、Webアプリケーションおよびネイティブアプリケーションのそれぞれの利点を備えるため、通信端末用アプリケーションの主流となっている。例えば、業務用の通信端末に用いられるアプリケーションとしてもハイブリッドアプリケーションが広く用いられる。業務用のハイブリッドアプリケーションを作成するためには、業務を行う各組織において業務目的に適したアプリケーションの開発を行う必要がある。

【0004】ここで、アプリケーションを開発するためには、アプリケーションをインストールする通信端末のOSに適合した専用端末を準備し、当該OSに適したアプリケーション開発ツールを導入し、当該アプリケーション開発ツールにおけるプログラミング言語を用いたアプリケーション開発をする必要がある。このよう

に、アプリケーション開発のためには、専用端末の準備およびアプリケーション開発ツールの導入などのアプリケーション開発環境の整備，ならびにプログラミング言語などのスキル習得などの多様な初期投資が必要である。

【0005】プログラミング言語のスキル習得の負担を軽減するために、プログラム開発を支援するための技術が開発されている。例えば、プログラム開発をより簡易化するために、特許文献1のようなプログラム開発装置が提案されている。

【発明が解決しようとする課題】【0007】しかしながら、通信端末上で動作し、通信端末の動作に合わせて動作または画面表示するアプリケーションを作成するためには特許文献1に記載された技術に比べてはるかに高度なプログラミング技術が必要である。さらに、アプリケーションの用途に応じて多様な機能をアプリケーションに組み込む必要があるため、上記のような通信端末上で動作するアプリケーションの開発のために、アプリケーション開発者に費用および時間の面で大きな負担がかかる。

【0008】本発明は、上記実情に鑑み、アプリケーション開発者への開発負担を軽減することができるアプリケーション生成支援システムおよびアプリケーション生成支援プログラムを提供することを目的とする。

【課題を解決するための手段】【0009】本発明の一実施形態に係るアプリケーション生成支援システムは、通信端末の動作に関連するパラメータに応じてアプリケーションの動作を規定する設定ファイルを設定する設定部と、前記設定ファイルに基づいてアプリケーションパッケージを生成する生成部と、を有する。

【0020】前記設定ファイルは、前記通信端末に固有のネイティブ機能を前記通信端末に実行させるプラグインファイルを含んでもよい。

【0037】前記設定ファイルは、前記通信端末に固有のネイティブ機能を前記通信端末に実行させるプラグインファイルを含んでもよい。

【0043】本発明によれば、アプリケーション開発者への開発負担を軽減することができるアプリケーション生成支援システムおよびアプリケーション生成支援

プログラムを提供することができる。

【発明を実施するための形態】〈第1実施形態〉【0062】・・・端末通信部330は全地球測位システム(Global Positioning System; GPS)機能を有していてもよい。GPS機能は第2通信端末300のネイティブ機能の一つである。GPS機能を有効にするアプリケーションを起動すると、端末通信部330は上空の衛星からの信号を受信し、第2通信端末300の現在位置を知ることができる。

【0064】カメラ350は、ディスプレイ340と反対側に設けられている。つまり、ディスプレイ340が設けられた側を表面とすると、カメラ350は裏面に設けられている。なお、カメラ350とは別個のカメラが表面に設けられていてもよい。カメラ350は、レンズユニットおよび撮像素子を有している。カメラ350の機能は第2通信端末300のネイティブ機能の一つである。カメラ350の機能を有効にするアプリケーションを起動すると、レンズユニットを介して撮像素子で検知される映像がディスプレイ340に表示される。この映像が撮像素子で検知されている間にユーザによって作動指示が入力されると、端末制御部320は、その作動指示に応じて撮像素子で検知された映像の静止画像を取得する。

【0101】「LANDSCAPE__LEFT」、 「LANDSCAPE__RIGHT」は、第2通信端末300を左横、右横に回転した場合の画面制御を指定する項目である。これらの項目では、第2通信端末300の画面を左横、右横に回転したときに、第2通信端末300の画面上でアプリケーションを正面表示させるか否かをチェックの有無で設定される。

【0102】「PORTRAIT__UPSIDE__DOWN」は、第2通信端末300を上下逆さに回転した場合の画面制御を指定する項目である。この項目では、第2通信端末300の画面を上下逆さに回転したときに、第2通信端末300の画面上でアプリケーションを正面表示させるか否かをチェックの有無で設定される。

【0103】「APP__START__PAGE__URL」は、アプリケーションの

スタートページとして指定されるWebページのURLを指定する項目である。この項目では、アプリケーションを起動したときに表示されるWebページのURLを設定するための文字列が設定される。この項目では、URLを表示するために、『http://』や『https://』などの規定の形式の文字列が設定される。

【0104】「APPLICATION_PLIST」は、アプリケーション設定ファイルを指定する項目である。この項目では、アプリケーションに搭載したいアプリケーション設定ファイルを設定し、当該設定ファイルをサーバ100にアップロードする。「APPLICATION_PLIST」において設定されるアプリケーション設定ファイルは、第1通信端末200を操作する第1ユーザによって作成された設定ファイルであってもよく、他の第3ユーザによって作成された設定ファイルであってもよい。

【0109】図10は、本発明の一実施形態に係るアプリケーション生成支援システムのパラメータチェックのシーケンスを説明する図である。図10に示すように、ステップ521で受信されたパラメータのチェックが行われる（ステップ5231）。ステップ5231では、例えば、設定されたパラメータの形式が規定の形式に適合しているか否か、およびアプリケーション設定ファイルの形式が規定の形式に適合しているか否かなどの設定状態がチェックされる。図8および図9に示した例では、「APP_START_PAGE_URL」に入力されたURLが『http://』または『https://』の形式になっているか否かがチェックされる。同様に、「APPLICATION_PLIST」でアプリケーション設定ファイルが規定のファイル形式になっているか否かがチェックされる。また、詳細は後述するが、アプリケーションに搭載される機能として第2通信端末300のネイティブ機能に関連するプラグインファイルが規定のファイル形式になっているか否かがチェックされる。

【0115】図13は、本発明の一実施形態に係るアプリケーション生成支援シ

システムの設定ファイルの一例を説明する図である。図13の「アイコン指定」, 「アプリケーション名指定」, および「画面の回転制御指定」はOS設定ファイルに該当する。つまり, 図8および図9の「CFBUNDLE_DISPLAY_NAME」, 「INFO_CFBUNDLE_VERSION」, 「LANDSCAPE_LEFT」, 「PORTRAIT_UPSIDE_DOWN」, および「LANDSCAPE_RIGHT」はOS設定ファイルに該当する。これらのOS設定ファイルに係る動作は, 複数の第2通信端末300に共通する動作である。図13の「アドレスバー表示」, 「画面のコピー&ペースト制御」, 「通信プロトコル制限」, 「無操作タイムアウト指定」, 「シングルサインオン設定」, および「バックボタンの利用制限」はアプリケーション設定ファイルに該当する。つまり, 上記の項目は「APPLICATION_PLIST」の一例である。これらのアプリケーション設定ファイルに係る動作は, 第1通信端末200によって入力された第2通信端末300の動作であり, 複数の第2通信端末300に共通する動作であってもよく, 特定の第2通信端末300に固有の動作であってもよい。

【0116】換言すると, 本実施形態の設定ファイルは, OS設定ファイルおよびアプリケーション設定ファイルを含む。OS設定ファイルは, ビルドシェル150が受信したパラメータに関連し, 複数の第2通信端末300の動作に共通するアプリケーションの動作を規定する設定ファイルである。アプリケーション設定ファイルは, ビルドシェル150が受信したパラメータに関連し, 第1通信端末200を操作する第1ユーザによって入力された情報に基づくアプリケーションの動作を規定する設定ファイルである。なお, 図13に示すOS設定ファイルおよびアプリケーション設定ファイルは一例であり, 本発明の「APPLICATION_PLIST」を限定するものではない。

〈第3実施形態〉【0135】[サーバ100Bの機能構成] 本実施形態のアプリケーション生成支援システム10Bにおけるサーバ100Bの各機能の構成は, 第1実施形態のサーバ100の各機能の構成と同じである。したがって, 図5を参照

してサーバ100Bの機能部について説明する。アプリケーション生成支援システム10Bのビルドシェル150Bに含まれるファイル設定部155Bは、OS設定ファイルおよびアプリケーション設定ファイルに加えて、第2通信端末300Bのネイティブ機能に関連するプラグインファイルを設定する点において、第1実施形態のファイル設定部155とは相違する。ファイル設定部155Bは、パラメータ受信部151Bが受信したパラメータにプラグインファイルに関連するパラメータが設定されている場合に、OS設定ファイルやアプリケーション設定ファイルと同様にプラグインファイルを設定する。

【0136】ファイル設定部155Bは、図6に示すように、パラメータ確認部1551B、テンプレート関連付け部1553B、およびプログラム更新部1555Bを有する。パラメータ確認部1551Bは、第1実施形態のパラメータ確認部1551と同様の機能を有する。テンプレート関連付け部1553Bは、第1実施形態のテンプレート関連付け部1553と同様の機能に加え、プラグインファイル用のテンプレートをパラメータ確認部1551Bによって選択された他のテンプレートに関連付ける。プログラム更新部1555Bは、第1実施形態のプログラム更新部1555と同様の機能に加え、プラグインファイル用のテンプレートによって規定された動作の各々が、第2通信端末300にインストールされたアプリケーション上で実現可能となるように、当該テンプレートの一部を更新する。

【0137】上記のように、プラグインファイルは、例えば、カメラ、GPS、マイク、加速度センサなど、第2通信端末300Bの機器に固有のネイティブ機能をアプリケーションによって第2通信端末300Bに実行させるための設定ファイルである。プラグインファイルは、OS設定ファイルおよびアプリケーション設定ファイルと同様に、ビルドシェル150Bが受信したパラメータに関連する設定ファイルである。

【0139】[アプリケーション生成支援システム10Bの動作フロー] 図17～図21を用いて、本実施形態のアプリケーション生成支援システム10Bの各機能

ブロックの動作について、フローチャートおよびユーザに表示されるインターフェースの一例を用いて詳しく説明する。

【0140】まず、図17～図19を用いて、第1通信端末200Bによって設定されたパラメータに基づいてアプリケーションを生成し、当該アプリケーションをダウンロード可能に提供する動作について説明する。図17は、本発明の一実施形態に係るアプリケーション生成支援システムのアプリケーション生成動作を示すフローチャートである。図17のフローチャートは図7のフローチャートと類似しているが、図17のフローチャートでは、ステップ527Bとステップ529Bとの間にプラグインファイルを設定するステップ528Bが設けられている点において、図7のフローチャートとは相違する。また、ステップ528Bが設けられていることに伴い、ステップ511Bで第1通信端末200Bに提供されるインターフェース600Bが図8のインターフェース600とは相違する。以下の説明において、図7の動作フローと同じ動作については説明を省略し、図7の動作フローと異なる動作について説明する。

【0141】まず、図17のステップ511Bで第1通信端末200Bに提供されるパラメータ設定GUIについて説明する。図18は、本発明の一実施形態に係るアプリケーション生成支援システムのパラメータ入力動作におけるインターフェースである。図18に示すインターフェース600Bは図8のインターフェース600と類似しているが、図18のインターフェース600Bの項目名620Bでは「APPLICATION_PLIST」の下に「PLUGIN_FILES」が設けられている点において、図8のインターフェース600とは相違する。

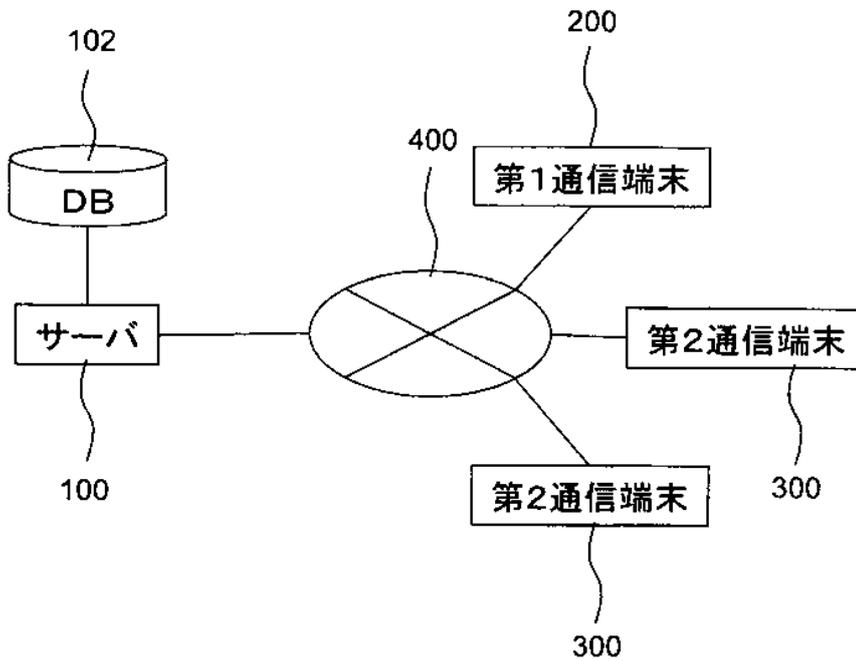
【0142】「PLUGIN_FILES」の入力欄630Bはプラグインファイル用のテンプレートが格納された位置を示すアドレスを受け付ける。この入力欄630Bには、ユーザが直接アドレスを入力してもよく、空白ボックスの右の「参照」をクリックすることで、参照するアドレスを指定してもよい。なお、プラグインファイル用のテンプレートは、指定されたアドレスに圧縮ファイル形式で格納さ

れている。ただし、当該ファイルは圧縮ファイル形式以外の形式で格納されていてもよい。

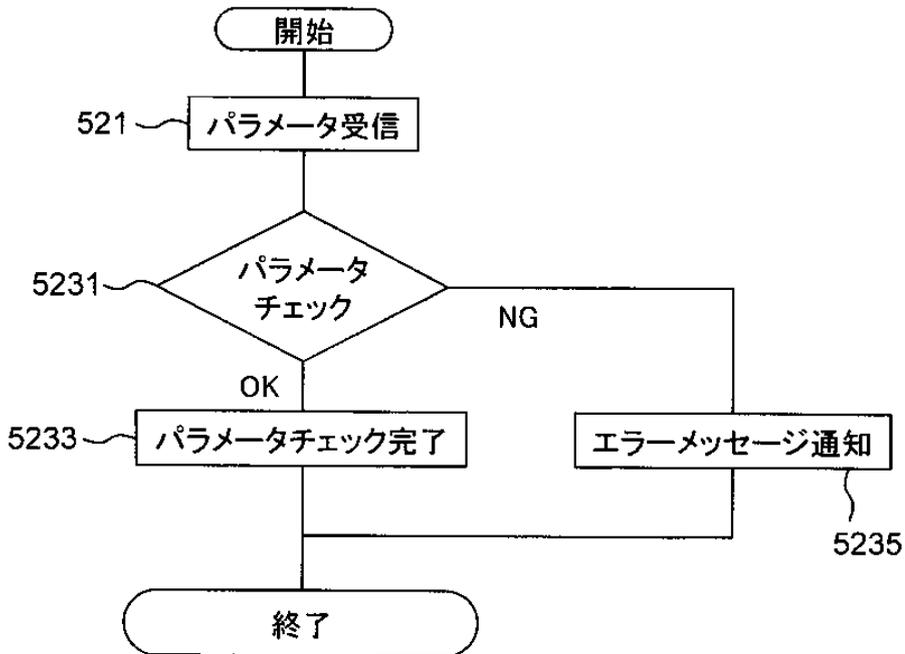
【0145】図17に示すように、ステップ527Bのアプリケーション設定ファイルの上書き処理に続いて、プラグインファイルの設定が行われる（ステップ528B）。プラグインファイルの設定は、OS設定ファイルの置換およびアプリケーション設定ファイルの上書き処理と同様に、設定ファイルを更新するという場合がある。ここで、ステップ528Bにおいて、プラグインファイルの設定するシーケンスについて図19を用いて説明する。

【図1】

10



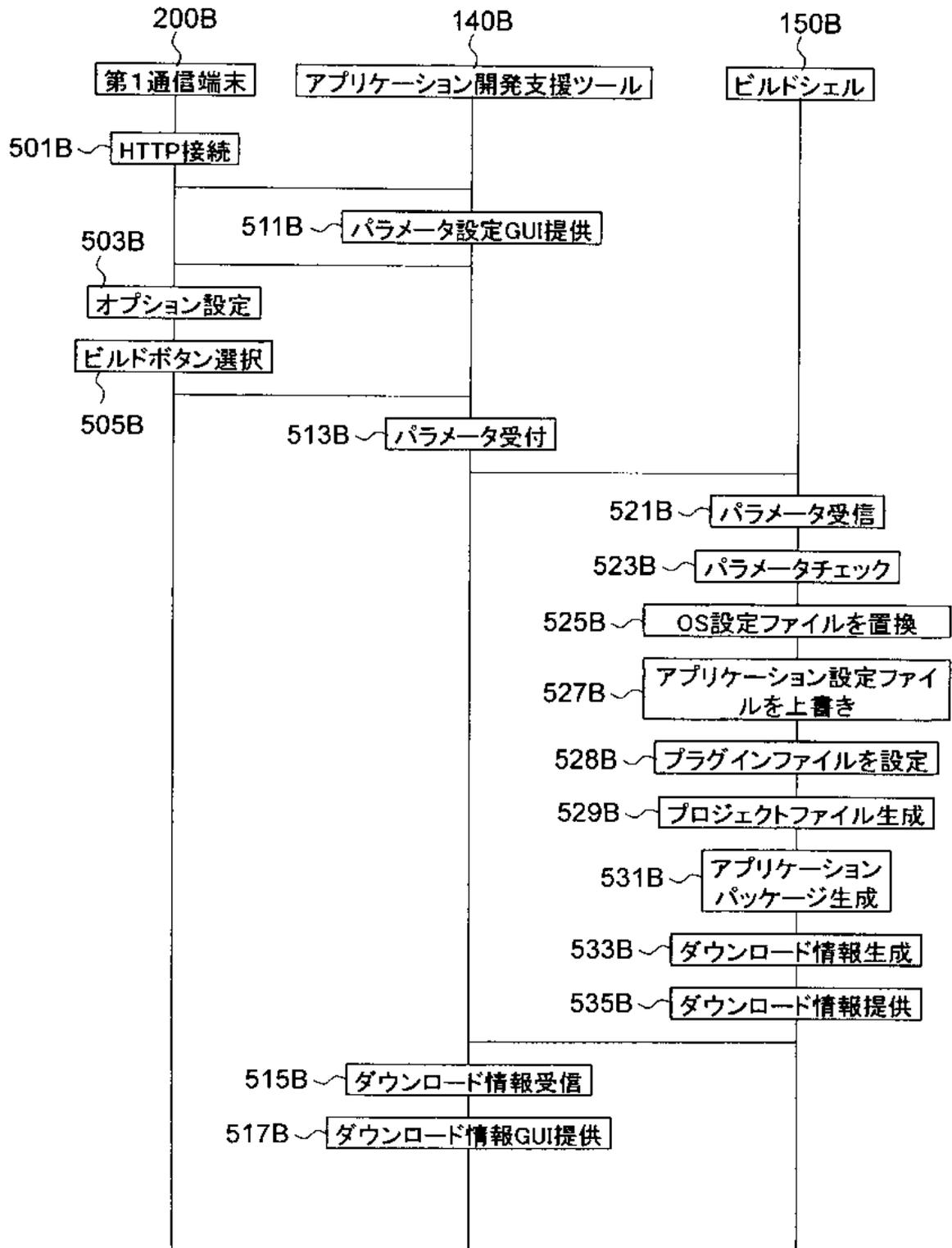
【図10】



【図13】

	設定ファイル 項目名	設定内容
OS設定ファイル	アイコン指定	アプリケーションのアイコン画像を指定する。
	アプリケーション名指定	アプリケーションのアイコン下に表示するアプリケーション名
	画面の回転制御指定	通信端末の縦横を変えた際の画面回転の制御
アプリケーション設定ファイル	アドレスバー表示	アドレスバーを表示するかどうかの設定
	画面のコピー&ペースト制御	画面に表示された内容のコピーおよび画面へのペーストの抑止有無
	通信プロトコル制限	HTTPやFTPなど通信プロトコルの制約設定
	無操作タイムアウト指定	一定時間アプリケーションを利用していない場合にアプリケーションを終了する設定
	シングルサインオン設定	異なるアプリケーション(プロセス)間での共通認証のシングルサインオンの有効化の設定
	バックボタンの利用制限	戻るボタンの利用制限の設定

【図17】



【図18】

600B

610B

プロジェクト XXX 01-PRODUCT-BUILD_1

このビルドはパラメータが必要です。

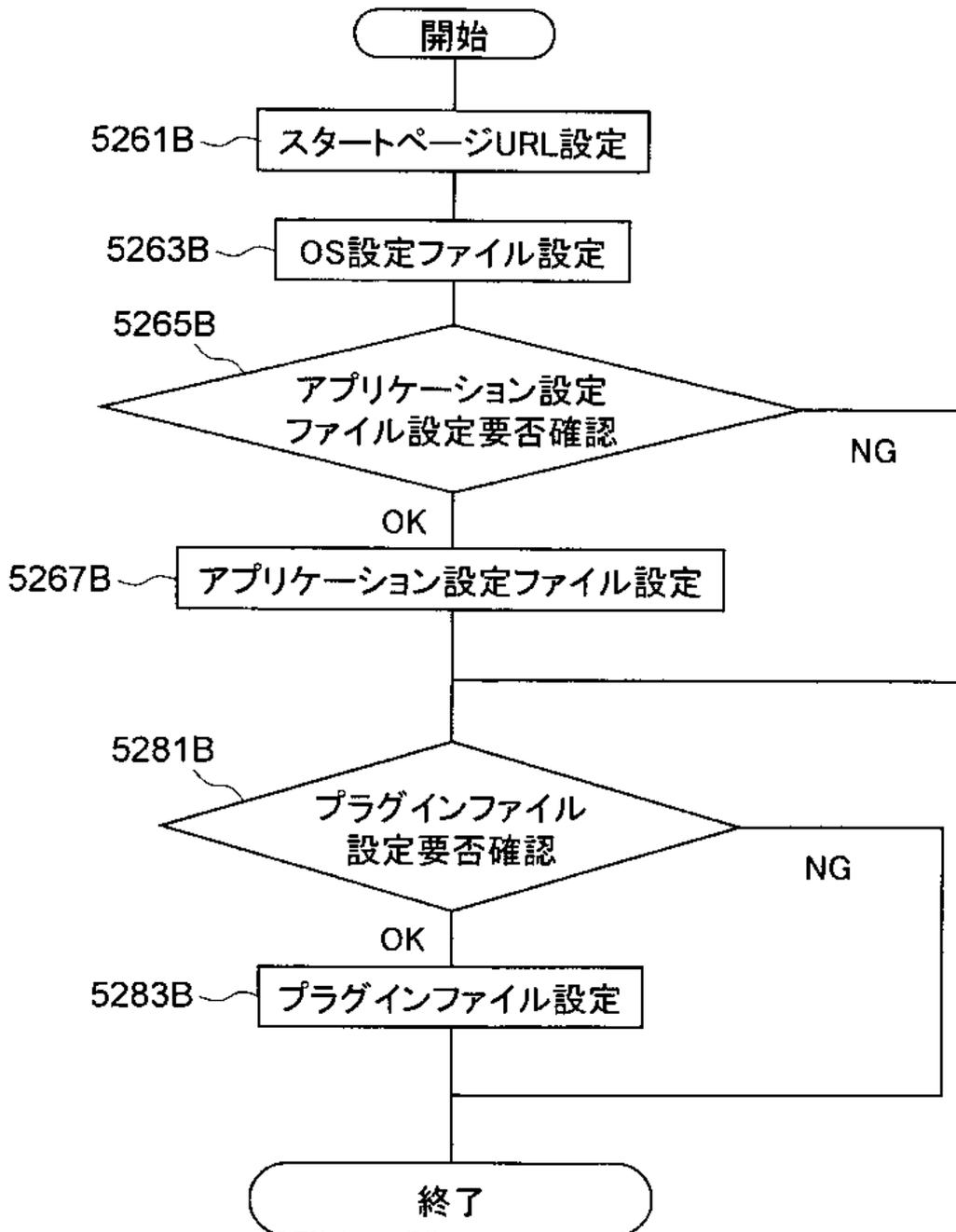
CFBUNDLE_DISPLAY_NAME	XXX管理	デバイス上のアプリケーションアイコン下に表示されるアプリケーション名を指定します。
INFO_CFBUNDLE_VERSION	1.1.1	アプリケーションのバージョンを指定する場合に入力します。
LANDSCAPE_LEFT	<input type="checkbox"/>	アプリケーションの画面制御の設定を行います。 左横に回転した場合に対応したいときはチェックを入れてください。
PORTRAIT_UPSIDE_DOWN	<input checked="" type="checkbox"/>	アプリケーションの画面制御の設定を行います。 上下逆さに回転した場合に対応したいときはチェックを入れてください。
LANDSCAPE_RIGHT	<input type="checkbox"/>	アプリケーションの画面制御の設定を行います。 右横に回転した場合に対応したいときはチェックを入れてください。
APP_START_PAGE_URL	https://XXX123.websystem..jp/xxx	アプリケーション起動時に表示されるURLの設定を行います。 「http://」もしくは「https://」から始まる文字列を入力してください。
APPLICATION_PLIST	<input type="text"/> 参照...	適用したいアプリケーション設定ファイルを指定します。 application.plistをアップロードしてください。
PLUGIN_FILES	Browser_camera 参照...	適用したいネイティブプラグインファイルを指定します。 zip形式のファイルをアップロードしてください。 日本語ファイル名には対応していません。

620B

630B

ビルド 640B

【図19】



2 (1) 引用文献1には、以下のとおりの記載がある（甲1）。

【技術分野】【0001】本発明は、アプリケーション生成装置、アプリケーション生成システム及びアプリケーション生成方法に関する。

【背景技術】【0002】近年、コンテンツマネジメントシステム（以下、CMS（Content Management System）という）によりウェブアプリケーションとして公開するコンテンツを構築し、管理することが行われている（例えば、特許文献1参照）。

【発明が解決しようとする課題】【0004】ところで、近年、ネイティブアプリケーションをダウンロードしてインストールすることができるスマートフォンが普及している。スマートフォンのユーザは、ネイティブアプリケーションをインストールする場合、アプリケーションを提供する所定のアプリケーションサーバにアクセスし、所望のネイティブアプリケーションを検索する。

【0005】しかしながら、CMSによって構築されるウェブアプリケーションは、ウェブサイトとして構築されるため、検索サイトの検索結果として表示されることがあるものの、アプリケーションサーバから検索することができない。したがって、アプリケーションサーバにおいてネイティブアプリケーションを検索したユーザに、CMSにより開発したウェブアプリケーションを利用してもらえないという問題がある。

【0006】これに対して、CMSによって構築したウェブアプリケーションと同等の機能を有するネイティブアプリケーションを開発し、当該ネイティブアプリケーションをアプリケーションサーバにアップロードすることも考えられる。しかしながら、ネイティブアプリケーションを新規に開発するには、多大な開発工数が必要であった。

【0007】本発明は、ネイティブアプリケーションを容易に生成することができるアプリケーション生成装置、アプリケーション生成システム及びアプリケーション生成方法を提供することを目的とする。

【課題を解決するための手段】【0008】本発明の第1の態様に係るアプリケーション生成装置は、端末にインストールされると、所定のアクセス先にアクセスして情報を取得し、前記端末の表示部に表示させるネイティブアプリケーションを生成

するアプリケーション生成装置であって、前記ネイティブアプリケーションのテンプレートであるテンプレートアプリケーションを記憶する記憶部と、ウェブアプリケーションのアドレスとともに、ネイティブアプリケーションの生成要求を示すリクエストを受け付ける受付部と、前記受付部が前記リクエストを受け付けたことに応じて、前記テンプレートアプリケーションの前記アクセス先を前記受付部が受け付けた前記アドレスに設定することで前記ネイティブアプリケーションを生成する生成部と、を備えることを特徴とする。

【0009】この発明によれば、アプリケーション生成装置は、生成部がテンプレートアプリケーションのアクセス先を、受付部が受け付けたウェブアプリケーションのアドレスに設定することで、当該テンプレートアプリケーションを、ウェブアプリケーションが表示する情報を表示するネイティブアプリケーションとして機能させることができる。したがって、アプリケーション生成装置は、受付部がウェブアプリケーションのアドレスを受け付けることで、当該ウェブアプリケーションと同等の機能を有するネイティブアプリケーションを容易に生成することができる。

【0010】また、前記テンプレートアプリケーションは、前記アクセス先を設定する設定情報を含み、前記生成部は、前記受付部が前記リクエストを受け付けたことに応じて、前記テンプレートアプリケーションをコピーし、コピーされた前記テンプレートアプリケーションに含まれる前記設定情報の前記アクセス先を前記受付部が受け付けた前記アドレスに設定することで前記ネイティブアプリケーションを生成することを特徴とする。

この発明によれば、アプリケーション生成装置は、設定情報に含まれているアクセス先を、受付部が受け付けたアドレスに変更するという単純な処理によってネイティブアプリケーションを容易に生成することができる。

【発明を実施するための形態】

【0022】以下、本発明の実施形態について、図面を参照しながら説明する。

<第1の実施形態>

[アプリケーション生成システムSの概要]

図1は、第1の実施形態に係るアプリケーション生成システムSの概要を示す図である。アプリケーション生成システムSは、アプリケーション生成装置1と、アプリケーション生成装置1とインターネット等のネットワークNを介して通信可能に接続された開発用端末2とを備え、ネイティブアプリケーションを生成するシステムである。

【0023】ネイティブアプリケーションは、例えば、スマートフォン等の携帯端末にインストールされるアプリケーションプログラムである。例えば、携帯端末のユーザは、アプリケーション提供サーバ3にアクセスして所望のネイティブアプリケーションを検索する。そして、携帯端末のユーザは、検索結果から所望のネイティブアプリケーションを選択し、アプリケーション提供サーバ3から当該ネイティブアプリケーションを取得して携帯端末にインストールする。

【0024】アプリケーション生成システムSのアプリケーション生成装置1は、例えば、開発用端末2の操作に応じて、ウェブアプリケーションを構築するコンテンツマネジメントシステムである。ここで、ウェブアプリケーションは、ネットワークNを介して実行されるアプリケーションであり、HTMLやJavaScript（登録商標）で記述される1以上のウェブページから構成されている。ウェブアプリケーションは、例えば、ブログ、有名人等のファンサイト、ゲームサイト、ショッピングサイト等である。

【0025】アプリケーション生成装置1は、例えば、ウェブアプリケーションのアドレスと、ネイティブアプリケーションの生成要求を示すリクエストとを受け付けると、受け付けたアドレスをアクセス先とするネイティブアプリケーションを生成して出力する。開発用端末2は、出力されたネイティブアプリケーションを取得し、ネットワークNを介して、アプリケーション提供サーバ3にアップロードする。これにより、携帯端末のユーザは、アプリケーション提供サーバ3において、ウェブアプリケーションの情報を表示するネイティブアプリケーションを検索すること

ができる。

【0026】その後、携帯端末のユーザが、当該ネイティブアプリケーションをアプリケーション提供サーバ3からダウンロードしてインストールを行うと、携帯端末の画面にアプリケーションのアイコンが表示される。携帯端末のユーザが、当該アイコンを実行するとネイティブアプリケーションが実行される。ネイティブアプリケーションは、起動すると、先に設定されたウェブアプリケーションのアドレスにアクセスしてウェブページを取得して表示する。そして、ネイティブアプリケーションは、表示されているウェブアプリケーションのリンク等が選択されると、当該リンク等にアクセスし、ウェブアプリケーションを構成する他のウェブページを取得して表示する。

【0027】[アプリケーション生成装置1の機能構成] 続いて、アプリケーション生成装置1の機能構成について説明する。図2は、第1の実施形態に係るアプリケーション生成装置1及び開発用端末2の機能構成図である。

アプリケーション生成装置1は、記憶部11と、受付部12と、生成部13と、変換部14と、送信部15とを備える。ここで、受付部12、生成部13、変換部14、及び送信部15は、例えば、CPU等により構成されている。

【0028】記憶部11は、例えば、ROM (Read Only Memory)、RAM (Random Access Memory)、ハードディスク等により構成される。記憶部11は、ネイティブアプリケーションのテンプレートであるテンプレートアプリケーション111を記憶する。

【0029】テンプレートアプリケーション111は、スマートフォン等の携帯端末にインストールされると、所定のアクセス先にアクセスして情報を取得し、当該端末の表示部に表示させるネイティブアプリケーションのテンプレートである。テンプレートアプリケーション111は、設定情報112と、1以上のプログラムファイル113とを含んでおり、記憶部11の所定のフォルダ内に格納されている。テンプレートアプリケーション111は、インストールされる端末のOSに対応し

て複数種類記憶されている。

【0030】設定情報112は、ネイティブアプリケーションのアクセス先、メニュー画面情報、及び表示態様情報に係る設定項目が含まれている。

メニュー画面情報に係る設定項目には、ネイティブアプリケーションのメニュー画面に用いられるメニュー画像と、ネイティブアプリケーションのメニュー画面に表示させるリンク先とが含まれている。

【0031】表示態様とは、背景色、フォントの形状及びアイコン画像の形状等であり、ネイティブアプリケーションの表示態様情報に係る設定項目には、背景色、フォント及びアイコン画像を選択する項目が含まれている。背景色は、ネイティブアプリケーションが端末の表示部に情報を表示させる際の背景色であり、アイコン画像は、ネイティブアプリケーションのアイコンを示す画像である。

【0032】1以上のプログラムファイル113は、例えば、汎用性の高い複数の関連プログラムをまとめたライブラリファイルや、設定情報112の各設定項目に基づいて情報を取得するためのプログラムファイル等から構成されている。

【0033】受付部12は、開発用端末2から、ウェブアプリケーションのアドレスとともに、ネイティブアプリケーションの生成要求を示すリクエストを受け付ける。具体的には、受付部12は、まず、開発用端末2から、リクエスト用ページの取得要求を受け付けると、リクエスト用ページを開発用端末2に送信する。図3は、第1の実施形態に係る開発用端末2の表示部23に表示されたリクエスト用ページ30の一例を示す図である。

【0034】図3に示すように、リクエスト用ページ30には、入力欄として、ネイティブアプリケーションがアクセスするウェブアプリケーションのロケーションを示すアドレスの入力欄31、ネイティブアプリケーションのメニュー画面に用いられるメニュー画像の入力欄32、及びネイティブアプリケーションのメニュー画面に表示されるリンク先の入力欄33が設けられている。ここで、入力欄32及び入力欄33は、複数設けられていてもよいし、当該リクエスト用ページに入力欄追

加ボタンを設け、当該ボタンが押下されたことに応じて追加表示されてもよい。また、リクエスト用ページ30には、ネイティブアプリケーションの表示態様情報に係る入力欄として、背景色の入力欄34、及びアイコン画像の入力欄35が設けられている。

【0035】例えば、入力欄31には、ウェブアプリケーションのメインページのURLが入力される。メニュー画像の入力欄32には、メニューページに表示するメニュー画像のファイルの格納先アドレスが入力され、リンク先の入力欄33には、当該メニュー画像に対応するリンク先のページのアドレスが入力される。

【0036】背景色の入力欄34には、カラーコード（RGB値を十六進法で表記した文字列）や背景画像を示すアドレスが入力される。アイコン画像の入力欄35には、画像ファイルのアドレスが入力される。ここで、入力欄32から入力欄35に入力する情報は、任意入力であってもよい。

【0037】リクエスト用ページ30には、入力欄の他に、ネイティブアプリケーションを出力するための画像である、出力ボタン36及び37が設けられている。

出力ボタン36は、第1種別のOS用のアプリケーションを生成して出力するためのボタンであり、出力ボタン37は、第2種別のOS用のアプリケーションを生成して出力するためのボタンである。なお、リクエスト用ページ30には、他の種別のOS用のアプリケーションを生成して出力するためのボタンが設けられていてもよい。

【0038】開発用端末2のユーザが、入力欄31から入力欄35に各種情報を入力し、ネイティブアプリケーションの出力ボタン36又は37を押下すると、開発用端末2は、ウェブアプリケーションのアドレスを含む各種情報と、ネイティブアプリケーションの生成要求を示すリクエストとを送信する。受付部12は、開発用端末2から、ウェブアプリケーションのメインページのアドレス、メニュー画面情報、表示態様情報及びOSの種別を示す情報とともに、ネイティブアプリケーションの生成要求を示すリクエストを受け付ける。

【0039】生成部13は、受付部12がネイティブアプリケーションの生成要求を示すリクエストを受け付けたことに応じて、テンプレートアプリケーションのアクセス先を受付部12が受け付けたウェブアプリケーションのアドレスに設定することでネイティブアプリケーションを生成する。

【0040】具体的には、生成部13は、受付部12がリクエストを受け付けたことに応じて、テンプレートアプリケーション111をコピーする。そして、生成部13は、リクエストとともに受信したOSの種別を示す情報に基づいて、複数のテンプレートアプリケーション111から、当該OSの種別に対応するテンプレートアプリケーション111を特定する。そして、生成部13は、特定したテンプレートアプリケーション111が格納されている所定のフォルダをコピーすることで、テンプレートアプリケーション111をコピーする。

【0041】そして、生成部13は、コピーして新たに生成されたテンプレートアプリケーション111に含まれる設定情報の内容を書き換え、ネイティブアプリケーションを生成する。生成されたネイティブアプリケーションは、携帯端末にインストールされて実行されると、設定情報に含まれているウェブアプリケーションのアドレスに基づいて、ウェブアプリケーションに対応するウェブページを取得する。そして、ネイティブアプリケーションは、設定情報に含まれている表示態様情報に基づいて取得したウェブページの表示態様を変更し、携帯端末の表示部に表示させる。

【0042】より具体的には、生成部13は、コピーされたテンプレートアプリケーション111に含まれる設定情報の設定項目「ネイティブアプリケーションのアクセス先」に、受付部12が受け付けた、ウェブアプリケーションのアドレスを設定する。

また、生成部13は、設定情報の設定項目「表示態様情報」に、受付部12が受け付けた表示態様情報（背景色、フォント、アイコン画像）を設定する。

【0043】また、生成部13は、受付部12が受け付けたメニュー画像、リンク

先のアドレスに基づいて、当該リンク先のアドレスに関連付けられたメニュー画像を複数含むメニューページを生成する。生成部13は、このメニューページを、ネイティブアプリケーションが実行された際に最初に表示されるページとしてもよい。

【0044】また、生成部13は、受付部12が受け付けた、ウェブアプリケーションのメインページのアドレスに基づいて、メインページを取得する。そして、生成部13は、取得したメインページに基づいて、ウェブアプリケーションのメニュー情報を生成する。例えば、生成部13は、取得したメインページに対応するウェブページのタグを解析し、重要度の高いリンクを抽出する。そして、生成部13は、抽出した重要度の高いリンクに含まれるメニュー情報を生成する。ここで、メニュー情報は、グローバルナビゲーションであり、ネイティブアプリケーションを介して表示される各ウェブページとともに表示される情報である。例えば、メニュー情報は、ネイティブアプリケーションを介して表示されたウェブアプリケーションのページの上部に付加される。端末のユーザは、ネイティブアプリケーションを介して表示されたウェブアプリケーションの各ページにおいて、グローバルナビゲーションに含まれる複数のリンクのいずれかを選択することにより、当該リンク先のウェブページに容易にアクセスすることができる。

【0045】変換部14は、生成部13が生成したネイティブアプリケーションを構成する設定情報と、1以上のプログラムファイルとを、端末がインストール可能な形式のファイルに変換する。具体的には、変換部14は、生成部13がネイティブアプリケーションを生成すると、当該ネイティブアプリケーションの設定情報及び1以上のプログラムファイルを格納している所定のフォルダを所定の方式で圧縮し、端末がインストール可能な形式のファイルに変換する。以下、変換したファイルをネイティブアプリケーションファイルともいう。

【0046】送信部15は、ネイティブアプリケーションファイルを所定の記憶領域に格納し、当該ファイルのアドレスを表示する送信用ページを開発用端末2に送信する。図4は、第1の実施形態に係る開発用端末2の表示部23に表示された送

信用ページ40の一例を示す図である。図4に示される送信用ページ40は、図3に示されるリクエスト用ページにおいて、第1種別のOS用のアプリケーションを出力するための出力ボタン36が押下されたときの表示例を示す。図4に示されるように、送信用ページ40には、第1種別のOS用のアプリケーションが生成された旨を示す情報と、当該第1種別のOS用のアプリケーションをダウンロードするための画像であるダウンロードボタン41とが表示されている。送信部15は、開発用端末2において、送信用ページ40に設けられているダウンロードボタン41が押下されると、生成したネイティブアプリケーションファイルを開発用端末2に送信する。

【0050】[処理フロー] 続いて、アプリケーション生成装置1の処理の流れについて説明する。図5は、第1の実施形態に係るアプリケーション生成装置1において、ネイティブアプリケーションが開発用端末2に送信されるまでの処理の流れの一例を示すフローチャートである。

【0051】まず、受付部12は、開発用端末2からリクエストを受け付ける(S1)。具体的には、要求部21が、アプリケーション生成装置1にリクエスト用ページ30の取得要求を行うと、受付部12は、開発用端末2に対してリクエスト用ページを送信し、開発用端末2から、ウェブアプリケーションのメインページのアドレス、メニュー画面情報、表示態様情報、OSの種別、及びネイティブアプリケーションの生成要求を示すリクエストを受け付ける。

【0052】続いて、生成部13は、受付部12がリクエストを受け付けたことに応じて、記憶部11の所定のフォルダに記憶されているテンプレートアプリケーション111をコピーする(S2)。ここで、生成部13は、S1において受け付けたOSの種別に対応したテンプレートアプリケーション111をコピーする。

【0053】続いて、生成部13は、S1において受け付けたウェブアプリケーションのメインページのアドレス、メニュー画面情報、表示態様情報に基づいて、S2においてコピーしたテンプレートアプリケーション111を構成する設定情報の

内容を書き換える（S3）。これにより、コピーされたテンプレートアプリケーション111は、ネイティブアプリケーションになる。

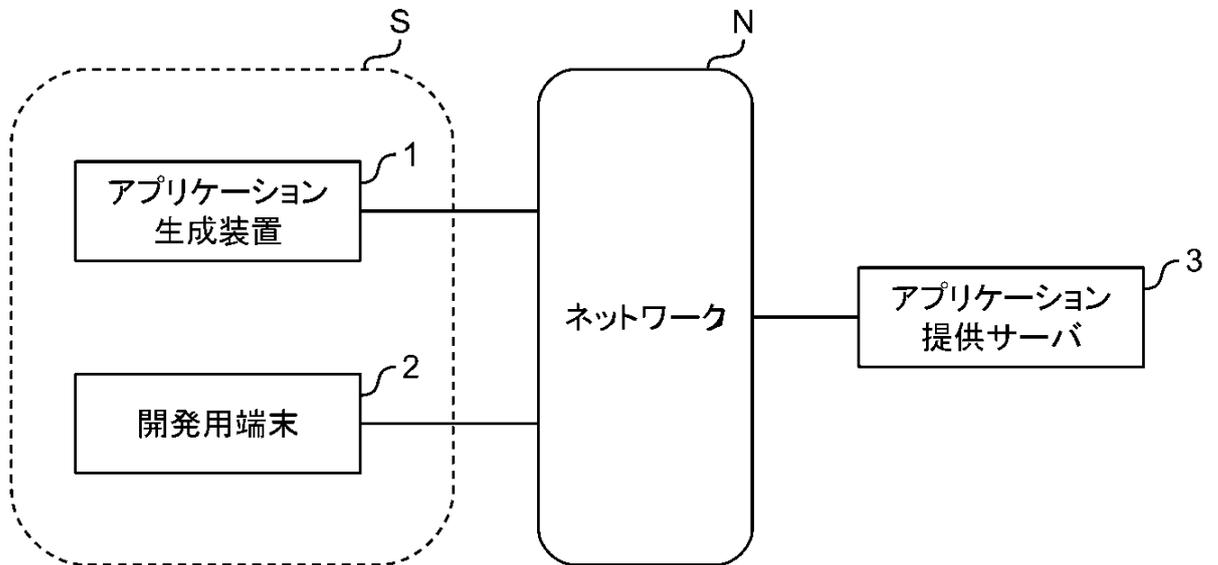
【0054】続いて、変換部14は、S3において生成部13が生成したネイティブアプリケーションの設定情報及び1以上のプログラムファイルを格納している所定のフォルダを所定の方式で圧縮し、ネイティブアプリケーションファイルに変換する（S4）。

【0055】続いて、送信部15は、ネイティブアプリケーションを開発用端末2に送信する（S5）。具体的には、送信部15は、ネイティブアプリケーションファイルの格納先のアドレス及びダウンロードボタンを表示する送信用ページを開発用端末2に送信し、開発用端末2において、当該ダウンロードボタンが押下されると、ネイティブアプリケーションファイルを開発用端末2に送信する。

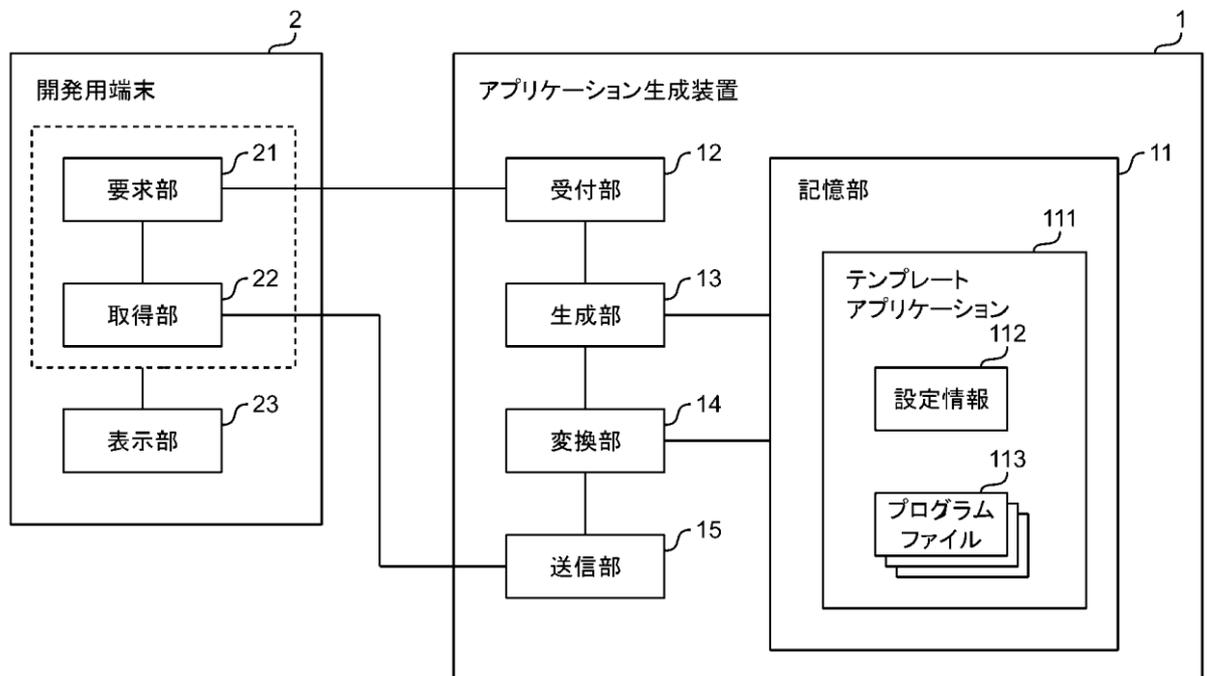
【0056】[第1の実施形態における効果]以上、第1の実施形態によれば、アプリケーション生成装置1は、記憶部11が、ネイティブアプリケーションのテンプレートであるテンプレートアプリケーション111を記憶し、受付部12が、ウェブアプリケーションのアドレスとともに、ネイティブアプリケーションの生成要求を示すリクエストを受け付け、生成部13が、リクエストを受け付けたことに応じて、テンプレートアプリケーション111のアクセス先を受付部12が受け付けたアドレスに設定することでネイティブアプリケーションを生成する。

【0057】このようにすることで、アプリケーション生成装置1は、生成部13がテンプレートアプリケーション111のアクセス先を、受付部12が受け付けたウェブアプリケーションのアドレスに設定することで、当該テンプレートアプリケーションを、ウェブアプリケーションが表示する情報を表示するネイティブアプリケーションとして機能させることができる。したがって、アプリケーション生成装置1は、受付部12がウェブアプリケーションのアドレスを受け付けることで、当該ウェブアプリケーションと同等の機能を有するネイティブアプリケーションを容易に生成することができる。

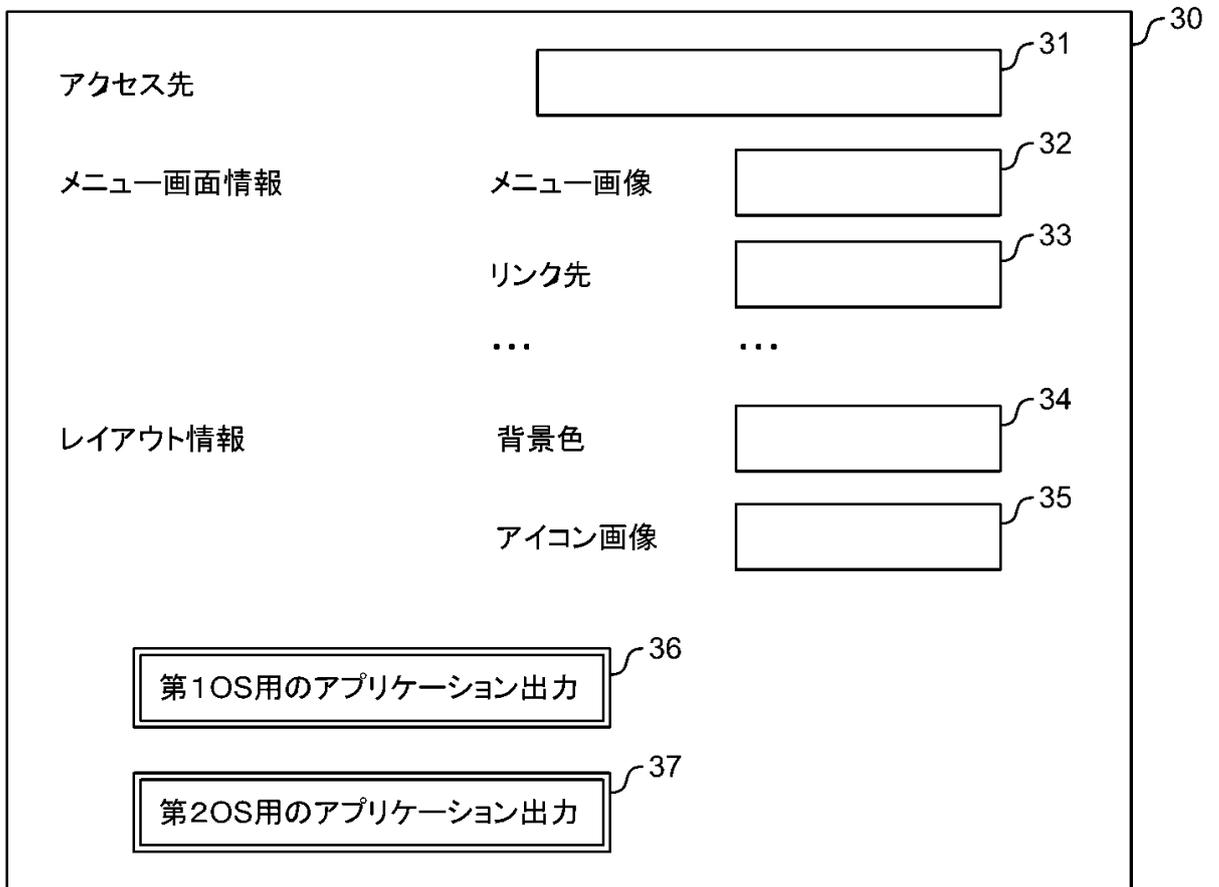
【図1】



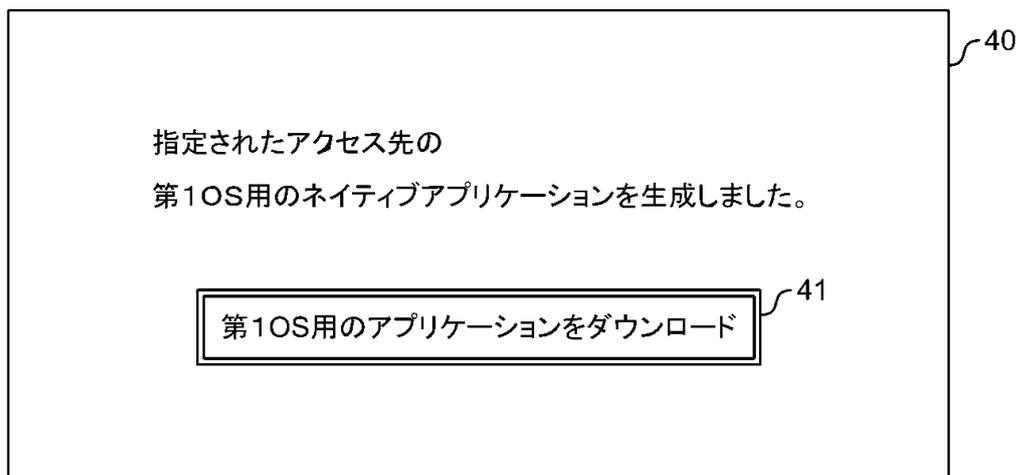
【図2】



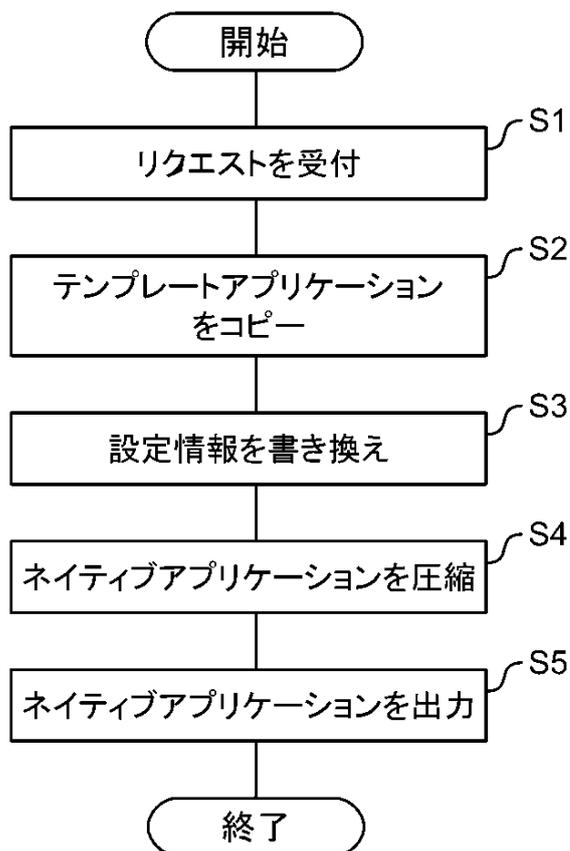
【図3】



【図4】



【図5】



(2) 引用文献2には、以下のとおりの記載がある（甲2）。

ア 「7. 2 カメラの利用

カメラは、アプリケーション制作者がもっとも使いたいと思うハードウェアの一つでしょう。カメラの機能は、単にその機能呼び出すだけでなく、取り出される映像の処理まで含めた知識が必要です。ここでその基本的な処理の仕方を身につけていきましょう。

7. 2. 1 カメラの利用とパーミッション

カメラも、アプリケーション内から利用できると応用範囲がグッと広がるハードウェア機能です。単に写真を撮影するだけでなく、AR（拡張現実）などのアプリにもカメラは利用されています。カメラ機能を利用したアプリというのは意外に多いのです。

このカメラ機能は、もちろんアプリケーション内から利用することができます。ただし、単に必要なソースコードを書けばOKというわけではないので注意が必要です。

カメラを利用するためには、アプリケーションに「パーミッション(アクセス権)」を用意する必要があります。これは、`AndroidManifest.xml`に記述します。`AndroidManifest.xml`を開き、`<manifest>`タグ内の適当なところに以下のタグを追記してください(よくわからなければ、最初の`<manifest>`の後を改行して記述してください)。

リスト7. 4

```
<uses-permission android:name="android.permission.CAMERA"/>
```

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```

```
<uses-feature android:name="android.hardware.camera"/>
```

これがカメラを利用する上で最低限必要なタグです。これらはそれぞれ以下のような役割を果たしています。

■`<uses-permission>`タグ

アプリケーションが必要とする許可を指定するものです。これはアプリケーションをインストールする際にチェックされます。

アプリケーションをインストールする際、「以下のアクセス許可を必要としています」といった表示が現れることがあります。あれは、実はこの`<uses-permission>`タグで記述されているものなのです。これにより、インストール時にアプリケーションが必要とするアクセス許可が表示され、ユーザーがそれを許可するとインストールされるようになります。

機能によっては、ユーザーが「これは勝手に使われたくないな」と思うようなこともあります。そこで、この<uses-permission>タグにより「このアプリはこれこれの重要な機能を使いますが、許可しますか?」ということを読み、ユーザーがそれを許可しないとインストールできないようにしているのです。

■<uses-feature>タグ

アプリケーションで必要とする機能を指定するものです。<uses-permission>タグと異なり、これはユーザーが許可をするようなものではありません。これは、アプリケーションへのインストールが可能かどうかを決めるのに利用されます。

例えば、この<uses-feature>でカメラを指定してあれば、そのアプリはカメラ機能がない端末にはインストールできなくなります。つまり「これこれの機能が用意されている端末でないと動かない」ということを指定するためのものなのです。

ここでは、<uses-permission>にandroid.permission.CAMERA、<uses-feature>にandroid.hardware.cameraを指定しています。これにより、このアプリはカメラが搭載されている端末でのみ動作すること、カメラ機能を利用するための許可が必要であることが指定されていたのです。

また、WRITE_EXTERNAL_STORAGEというのは、外部ストレージへの書き出しを許可するものです。イメージの保存を行うため、これも用意しておきましょう。」(356頁15行~357頁31行)

イ 「7.3.3 GPS利用のパーミッション設定

では、実際にGPSを利用したサンプルを作ってみましょう。まず最初に行うのは、AndroidManifest.xmlの変更です。

GPSも、やはりパーミッション関係の情報を用意しなければ利用できないようになっています。AndroidManifest.xmlを開き、<manif

est>タグ内に以下のタグを追加してください。

リスト7.8

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-feature android:name="android.hardware.Location"/>
<uses-feature android:name="android.hardware.Location.gps"/>
<uses-feature android:name="android.hardware.Location.network"/>
```

最初の<uses-permission>では、ACCESS_FINE_LOCATIONへのパーミッションを設定しています。これにより、ユーザーが位置情報へのアクセスを許可しないとアプリがインストールできないようになります。

残る3つのタグは、<uses-feature>タグです。すなわち、これらの機能を内蔵するAndroid端末にしかインストールできないようにするものです。これにより、GPSやネットワーク機能がない端末にはアプリケーションをインストールできないようになります。

これらのタグは、GPS利用には必須のものと考え、必ず用意してください。」(368頁11行～29行)

(3) 引用文献3には、以下のとおりの記載がある(甲3, 乙1)。

ア 「第7章 HTML5+JavaScriptでアプリ開発
はじめてのPhoneGap(その1)

ここまでできる

PhoneGap」(66頁表題)

イ 「PhoneGapを利用すると、iPhoneやAndroid, Windows Phoneなどのマルチプラットフォームに対応するアプリを作成でき

ます。特にGPSなどのネイティブ機能にアクセスできる点が注目されています。」

(66頁副題)

ウ 「現在同じコードから複数のプラットフォームにアプリを生成できる表1のようなフレームワークがあります。

ここでは、数あるフレームワークの中からPhoneGapを紹介します(図1).」

(66頁中欄下から2行～右欄3行)

表1 同一コードから複数のプラットフォームにアプリを生成できるフレームワーク

フレームワーク名	言語	概要	URL
PhoneGap	HTML, CSS, JavaScript	iOS, Android, Windows Phone7など多数のプラットフォーム向け。実体はブラウザUI上で動くアプリとなる	http://phonegap.com/
Titanium Mobile	JavaScript	iOS, Android向けJavaScriptからネイティブアプリを生成	http://www.appcelerator.com/products/titanium-mobile-application-development/
Adobe AIR	ActionScript, HTML, CSS, JavaScript	ActionScriptまたはJavaScriptを使ってアプリを作る	http://www.adobe.com/jp/products/air.html
Rhodes	Ruby	Ruby on Railsをベースとした開発 iOS, Android, WindowsPhone7など多数のプラットフォーム向け	http://rhone.com/products/rhodes/

エ 「PhoneGapでできること

PhoneGapでできることは、以下の通りです。

①HTML5+CSS+JavaScriptを使用してネイティブアプリを作成

②プラグインの仕組みを使って、カメラやGPSなど端末のネイティブ部分にアクセス可能

③iOS, Android, BlackBerry, WebOS, Windows Phone 7, Symbian, Bada向けにアプリを作成できる

①から分かるように、アプリを作るために必要な知識は、Webに関係する技術のみになります。

もっとも注目すべき特徴は②のプラグインの仕組みがあることです。通常、HTML5やJavaScriptだけでは、スマートフォンで実現できないことがた

くさんあります。例えば、AndroidのIntent機能を使うにはネイティブの実装が必要です。そういった部分をプラグインがカバーすることで、ネイティブの機能を使ってより良いアプリを作ることができます。」(66頁右欄13行～67頁左欄13行)

オ 「PhoneGapを使って、
地図とGPSを組み合わせる

PhoneGapのすばらしさを体験するのに、地図とGPSを組み合わせたアプリを例に紹介します。

ここでは、Google Maps JavaScript API V3 (<http://>以下略) を使ってスマートフォン上に地図を表示します。さらに、PhoneGapの「Geolocation」を使用して、現在位置にマーカを表示します。

リスト4のソースを用意し、先ほど作成したプロジェクトのindex.htmlと置き換えます。マーカ用の20ピクセルの画像をpin.pngという名前で用意します。

各プラットフォームでビルドし、実行すると、iPhone (図14)、Android (図15)、Windows Phone (図16) のように現在値の地図と、現在値のマーカが表示されます。また、地図をタップするとタップした場所にマーカが表示されます。



図14 iPhoneで実行



図15 Androidで実行

●仕組み

作成したリスト4を解説します。

<script type="text/javascript">の部分でGoogle Mapのjsファイルを読み込んでいます。パラメータsensorはGPSがある端末の場合は、trueを設定します。それ以外の場合は、falseを設定します。アプリのターゲットであるスマートフォンではGPSを搭載しているのでtrueと設定しています。」(70頁右欄6行～72頁左欄25行)

(4) 引用文献4には、以下のとおりの記載がある(甲4)。

ア 「ソースコード、リソース、設定、ライブラリ

...

設定ファイルは、そのアプリのアプリ名やバージョン、対応言語（英語や日本語 e t c .）などを記述しておくファイルです。中身は細かな項目が多いといえます。一見、そのような情報はソースコードに記述しておけば良いのではないかと思ってしまうのですが、設定ファイルとして独立させておくことで、i O S や A n d r o i d はそのアプリがどのような属性を持つアプリなのかをすぐに把握できるようになるわけです。」（10 頁右欄 2 行～11 頁左欄 6 行）

イ 「設定ファイル「i o s _ s a m p l e - I n f o . p l i s t」

(5)の i o s _ s a m p l e - I n f o . p l i s t はアプリの設定ファイルです。アプリの名前やバージョン、そのアプリが利用する S t o r y b o a r d のファイル等の情報が記述されています。i O S は (5) を読み取ることで、それらの情報を取得します。」（14 頁左欄下から 2 行～右欄 4 行）

ウ 「超重要な設定ファイル「A n d r o i d M a n i f e s t . x m l」

(5)の A n d r o i d M a n i f e s t . x m l は、頻繁に編集することになる超重要な設定ファイルです（図 14）。この中ではアプリの名前やアイコン、対応する A n d r o i d のバージョン、起動時に実行する O O A c t i v i t y クラス、そのアプリをネットワークに接続できるようにするか？、SDカードにファイルを保存できるようにするか？などの、様々な項目を設定します。」（16 頁右欄下から 8 行～末行）

(5) 引用文献 5 には、以下のとおりの記載がある（甲 5，乙 2）。

ア 「J a v a S c r i p t / H T M L 5 + P h o n e G a p の組み合わせが一番

スマホアプリの作り方といっても、実はいろいろな方法があります（図 1）。それらはいずれも一長一短で、快適に動くアプリを作れるけれど作り方が難しい方法もあれば、簡単にアプリを作れるけれど使い勝手の良いアプリは作りづらい方法もあります。

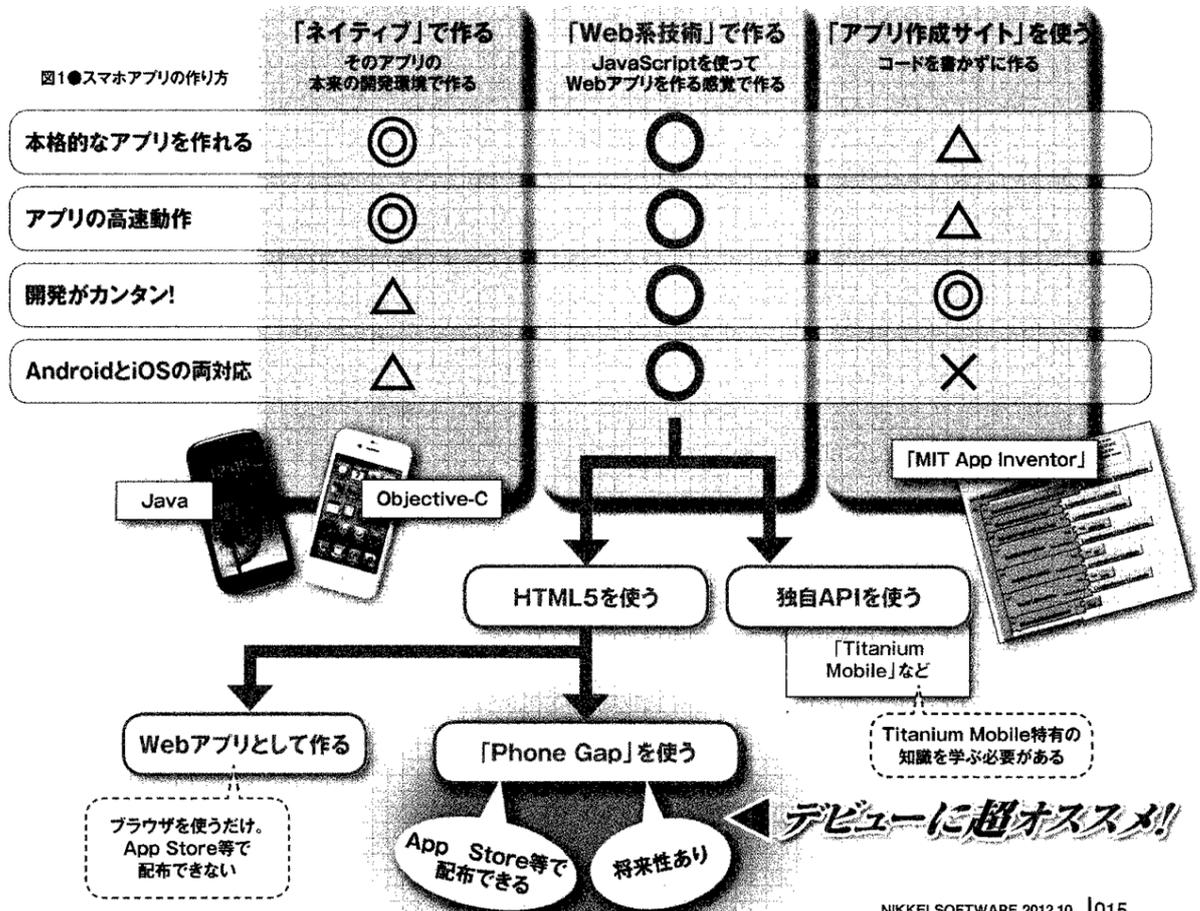
日経ソフトウェア編集部では、様々な方法を「ハードルが一番低そうなものは

どれか？」という点で吟味しました。その結果、この特集記事では「JavaScriptとHTML5+PhoneGapの組み合わせ」を採用することにしました。

JavaScriptとHTML5は、主にWebページで使われるテクノロジーです。Webブラウザとテキストエディタさえあればすぐにプログラミングができるので、ハードルは低いと言えるでしょう。

PhoneGap（フォンギャップ）はJavaScriptとHTML5のプログラムをスマホアプリ化する無償のソフトウェアです。Android版もiOS版もあります。

要するに、JavaScriptとHTML5+PhoneGapは、最もとっつきやすく、それでいてちゃんとしたアプリを作れる組み合わせなのです。」（16頁左欄7行～27行）



イ 「ステップ4

横画面に固定する

スマートフォンでは端末が縦の姿勢のときと、横の姿勢のときで画面が切り替わります。ただ、ゲームでは画面が切り替わらない方が望ましいので、ここでは横画面に固定する設定を施します。

Androidでは、`AndroidManifest.xml`に「`android:screenOrientation = "landscape"`」の1行を追加します (図5)。

iOSでは、左側のリストの青いアイコンのプロジェクト名 (Matatabi Pakkun) を選択すると、その右側に「TARGETS」という項目が現れるので、その下にある「Matatabi Pakkun」を選びます。すると、プロジェクトの各種設定が表示されます。この中に「Supported Device Orientations」という項目があるので、「Landscape Left」か「Landscape Right」を選びます。

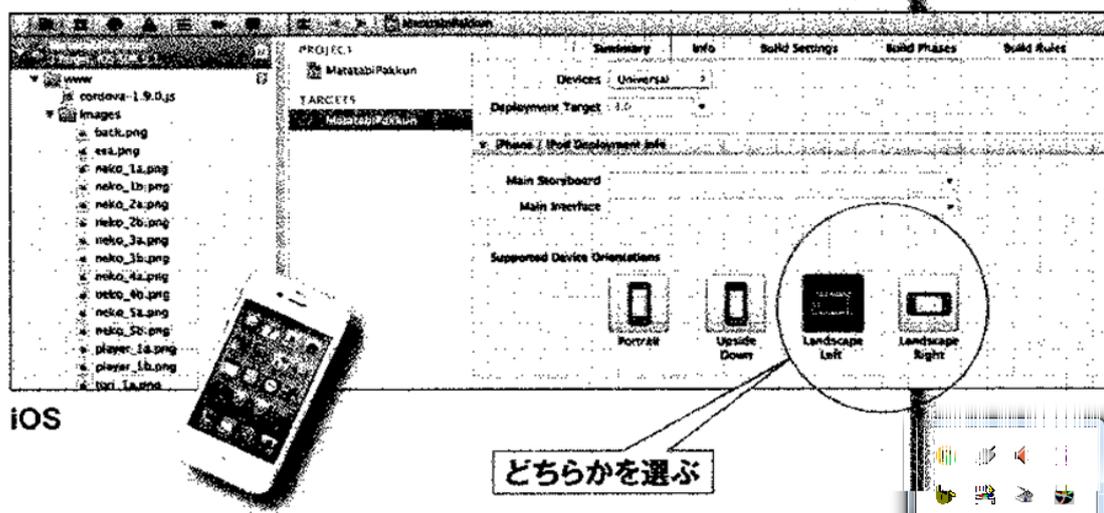
これで、画面を横向きに固定できます。なお、ここでいうorientation は「方向」、landscapeは「横方向」という意味です。」(29頁左欄1～16行)

図5●横画面に固定する設定

```
<activity
  android:name=".MatatabiPakkunActivity"
  android:label="@string/title_activity_matatabi_pakkun"
  android:screenOrientation="landscape">
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
  </intent-filter>
</activity>
```

追加

Android



iOS

どちらかを選ぶ

ウ 「ステップ4

縦画面に固定する

この電子書籍アプリは縦画面に固定します。そこで図6の設定を施してください。Androidでは、AndroidManifest.xmlに「android:screenOrientation="portrait"」の1行を追加します。iOSでは、「Supported Device Orientations」の項目で「Portrait」を選びます。なお、portraitここでは「縦方向」の意味です。」(36頁左欄下から10行~末行)

Android

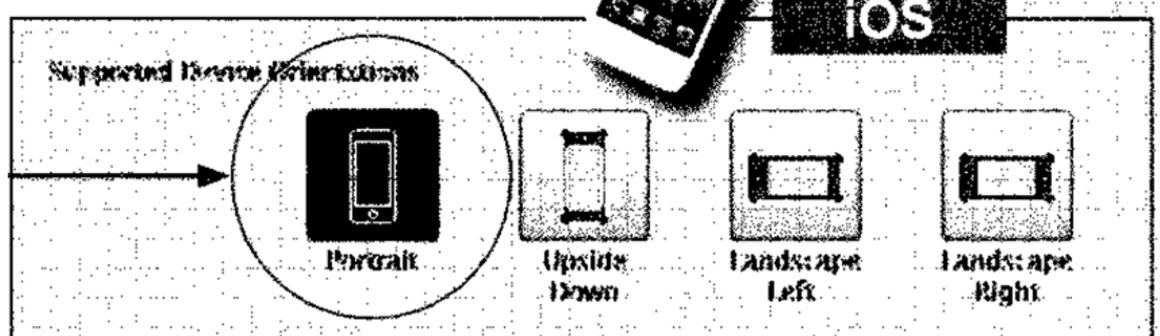
```

<activity
  android:name=".DigiBookActivity"
  android:label="@string/title_activity_digi_book"
  android:screenOrientation="portrait">
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
  </intent-filter>
</activity>

```

追加

図6●縦画面に固定



(6) 参考文献1には、以下のとおりの記載がある（甲8）。

ア 「3Dグラフィクスと

加速度センサ

今回のサンプルアプリケーションは、加速度センサとOpenGL ESを用いた3Dグラフィクス描画、それに、前のセクションで解説した音声再生を組み合わせたものになります（図1）。

サンプルアプリケーションを起動すると、3Dグラフィクス界ではおなじみのティーポットの3Dモデルが表示されます。端末を傾けると視点が変わり、見る向きを変えることができます。また、画面をタッチして左右に指を動かすと、ティーポットを回転させることができます。音声は画面をタップすると再生を開始し、再生している間はティーポットの色が変わります。

なお、ネイティブコードのみで実装する関係で、対象となるデバイスはGingerbread (Android 2.3), NDKはRevision 5以降が対象となります。

それでは、今回もすべてのソースコードを含むプロジェクトファイルは、「Software Design」2011年8月号の解説記事「Androidエンジニアからの招待状」のサポートページ (<http://以下略>) からダウンロードできますので、そちらをご覧くださいながら本稿を読み進めてもらえればと思います。

センサからの値の取得

デバイスに搭載されている各種センサの値をネイティブコードからも取得することができます。今回は表示するティーポットの向きを決めるために加速度センサを使って重力の方向を求めます。

センサに関するAPIと定数の定義はinclude/android/sensor.hにあります。」(157頁左欄1行～右欄末行)

イ 「■AndroidManifest.xml

加速度センサでデバイスの傾きを取得するため、傾けた際に画面のオリエンテーションが変わらないように、android:screenOrientation="portrait"を追加してportraitで固定します。

```
<activity android:name="
    android.app.NativeActivity"
    android:label="
    @string/app_name"
    android:screenOrientation="
    portrait">
```

(162頁左欄13行～23行)

(7) 乙3文献には、以下のとおりの記載がある(甲7, 乙3)。

ア 「PhoneGapは、HTML5, CSS3, JavaScript

などのWeb標準技術を用いてiOSやAndroidなどのスマートフォン向けのネイティブアプリを開発するためのフレームワークです。」(155頁左欄10行～14行)

イ 「カメラアプリの作成

ここからはHTMLファイルを1つ作成して、各OSで動作することを確認します。今回はカメラで写真を撮影（もしくは、すでにある写真から選択）して、写真を画面に表示するカメラアプリを作ってみます。

index.htmlファイルの作成

自分の好きなエディタを使って、リスト3のindex.htmlファイルを作成します。jQueryとjQuery Mobileのファイルを使用していますので、必要なファイルをjQueryのWebサイトから取得して展開しておいてください。

ポイントは、JavaScript部分のcapturePhoto () 関数とgetPhoto () 関数の中です。navigator.camera～とあるのは、PhoneGapのライブラリの関数を使用しています。

PhoneGap側では、ネイティブの対応する機能呼び出します。ここでは、カメラの撮影機能とギャラリー機能呼び出しています（実際に呼び出されるアプリは、OSによって異なります）。

これ以外にもJavaScriptでPhoneGapのAPIを呼び出せば、センサーやGPSの機能を使うことができます。

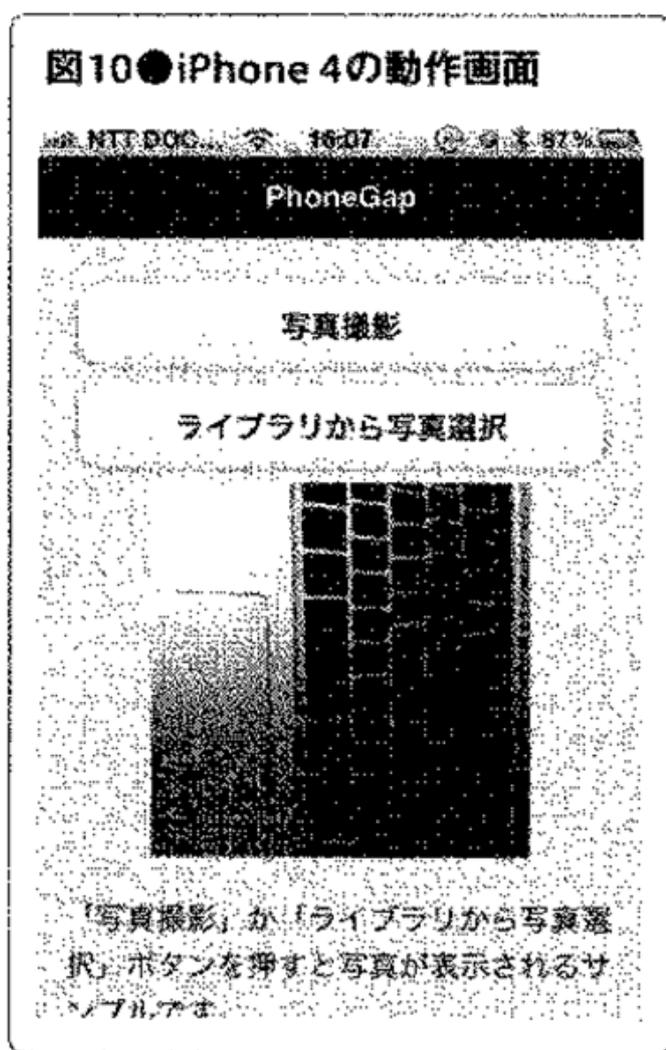
これでカメラアプリを作成する準備が整いました。

iOS編

作成したindex.htmlファイルやjQuery関連のファイルを、HelloWorld作成時に作ったXcodeのプロジェクトの中に入れます。wwwフォルダの下にはindex.htmlファイルがすでに存在していますので、作成したファイルで置き換えます。

置き換えたらXcodeの実行ボタンを押して起動してみましょう。図10のような画面が表示されれば成功です。エミュレータの場合カメラは起動しませんが、その際はエラーが出ていることを確認できるかと思います。

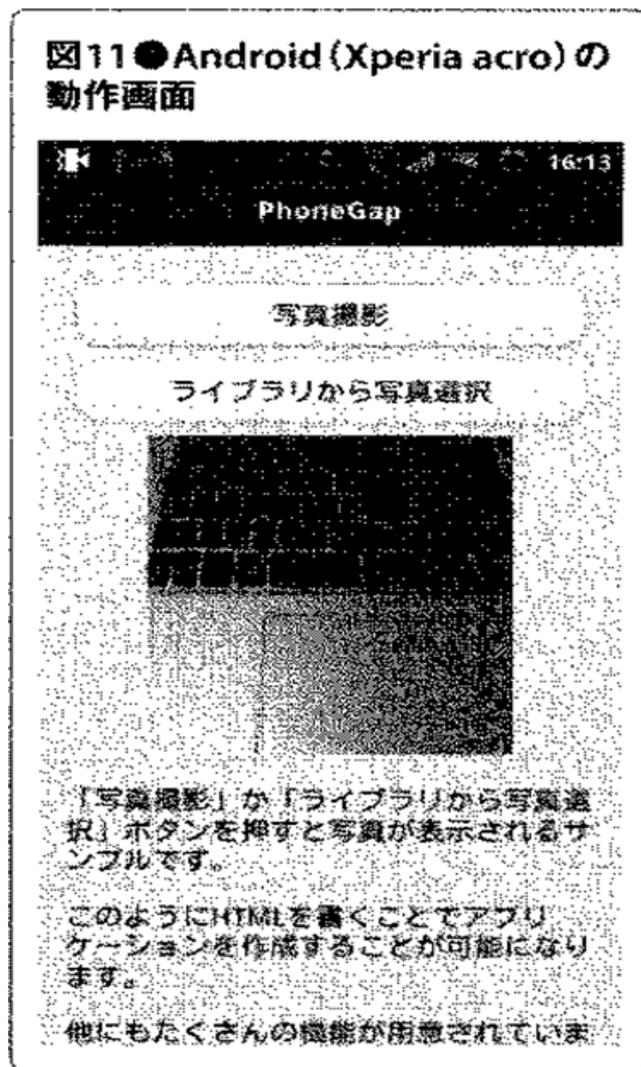
iPhoneやiPadで実行するとカメラで写真を撮って、画面に表示するのを確認できます。」(159頁左欄1行～右欄4行)



ウ 「Android編

Eclipseを起動して、先ほどのHello Worldプロジェクトのassets/wwwフォルダの中に「index.html」やjQuery関連

のファイルをコピーします。実行ボタンを押して実行すると、iOSの環境と同じように動いていることが確認できます（図11）。」（160頁右欄1行～7行）



3 取消事由1（独立特許要件違反の有無—本件補正発明の進歩性の有無）について

(1) 本件補正発明と引用発明との一致点及び相違点

ア 前記2(1)で認定した引用文献1の記載からすると、引用発明の構成は以下のとおり認められる。

「アプリケーション生成装置1と開発用端末2とがインターネット等のネットワ

ークNを介して通信可能に接続され、スマートフォン等の携帯端末にインストールされるアプリケーションプログラムであるネイティブアプリケーションを生成するアプリケーション生成システムであって、

アプリケーション生成装置1は、記憶部11と、受付部12と、生成部13と、変換部14と、送信部15とを備え、

記憶部11は、ネイティブアプリケーションのテンプレートであるテンプレートアプリケーション111を記憶し、

ここで、テンプレートアプリケーション111は、スマートフォン等の携帯端末にインストールされると、所定のアクセス先にアクセスして情報を取得し、当該端末の表示部に表示させるネイティブアプリケーションのテンプレートであり、

テンプレートアプリケーション111は、設定情報112と、1以上のプログラムファイル113とを含んでいて、記憶部11の所定のフォルダ内に格納されており、

受付部12は、開発用端末2から、リクエスト用ページ30の取得要求を受け付けると、リクエスト用ページ30を開発用端末2に送信し、

ここで、リクエスト用ページ30には、入力欄として、ネイティブアプリケーションがアクセスするウェブアプリケーションのロケーションを示すアドレスの入力欄31が設けられ、また、ネイティブアプリケーションの表示態様情報に係る入力欄として、背景色の入力欄34、及びアイコン画像の入力欄35が設けられており、

入力欄31には、ウェブアプリケーションのメインページのURLが入力され、背景色の入力欄34には、カラーコードや背景画像を示すアドレスが入力され、アイコン画像の入力欄35には、画像ファイルのアドレスが入力され、

受付部12が、開発用端末2から、ウェブアプリケーションのメインページのアドレス、及び表示態様情報とともに、ネイティブアプリケーションの生成要求を示すリクエストを受け付けると、生成部13は、コピーして新たに生成されたテンプレートアプリケーション111に含まれる設定情報の内容を、受付部12が受け付

けたウェブアプリケーションのメインページのアドレス、及び表示態様情報に基づいて書き換えてネイティブアプリケーションを生成し、

ここで、生成されたネイティブアプリケーションは、携帯端末にインストールされて実行されると、設定情報に含まれているウェブアプリケーションのアドレスに基づいて、ウェブアプリケーションに対応するウェブページを取得し、設定情報に含まれている表示態様情報に基づいて取得したウェブページの表示態様を変更して、携帯端末の表示部に表示させるものであり、

変換部14は、生成部13がネイティブアプリケーションを生成すると、当該ネイティブアプリケーションの設定情報及び1以上のプログラムファイルを格納している所定のフォルダを所定の方式で圧縮し、端末がインストール可能な形式のネイティブアプリケーションファイルに変換し、

送信部15は、ネイティブアプリケーションファイルを所定の記憶領域に格納し、当該ファイルのアドレスを表示する送信用ページを開発用端末2に送信し、開発用端末2において、送信用ページ40に設けられているダウンロードボタン41が押下されると、生成したネイティブアプリケーションファイルを開発用端末2に送信する、

アプリケーション生成システム。」

イ 前記第2の2(2)で認定した本件補正発明の構成と前記アの引用発明の構成を比較すると、両発明の一致点及び相違点は以下のとおりとなる。

(ア) 一致点

「携帯通信端末の所定の機能を実行させるためのパラメータに応じて、前記携帯通信端末において実行されるアプリケーションの動作を規定する設定ファイルを設定する設定部と、

前記設定ファイルに基づいてアプリケーションパッケージを生成する生成部と、を有するアプリケーション生成支援システム」である点。

(イ) 相違点

a 相違点1

設定ファイルを設定するパラメータが、

本件補正発明では、「携帯通信端末に固有のネイティブ機能を実行させるためのパラメータ」であるのに対して、

引用発明では、携帯通信端末の機能を実行させるためのパラメータではあるものの、携帯通信端末に固有のネイティブ機能を実行させるためのパラメータであることが特定されていない点。

b 相違点2

設定ファイルが、

本件補正発明では、「アプリケーションの、携帯通信端末の動きに伴う動作」を規定する「設定ファイル」であるのに対して、

引用発明では、「設定情報」が、ネイティブアプリケーションのウェブページ取得動作や表示動作を規定するものの、「携帯端末の動きに伴う」動作を規定するものであることが特定されていない点。

(2) 相違点1についての判断

本件審決は、引用発明に引用文献2～5及び参考文献1記載の技術（同技術に乙3文献記載の技術を併せて、以下「被告主張周知技術」という。）を適用することにより、本件補正発明に想到し得ると判断していることから、引用発明に被告主張周知技術を適用する動機付けの有無について検討する。

ア 引用発明

前記2(1)のとおり、引用文献1には、「近年、コンテンツマネジメントシステム（以下、CMS（Content Management System）という）によりウェブアプリケーションとして公開するコンテンツを構築し、管理することが行われている（例えば、特許文献1参照）。ところで、近年、ネイティブアプリケーションをダウンロードしてインストールすることができるスマートフォンが普及している。スマートフォンのユーザは、ネイティブアプリケーションをインストー

ルする場合、アプリケーションを提供する所定のアプリケーションサーバにアクセスし、所望のネイティブアプリケーションを検索する。しかしながら、CMSによって構築されるウェブアプリケーションは、ウェブサイトとして構築されるため、検索サイトの検索結果として表示されることがあるものの、アプリケーションサーバから検索することができない。したがって、アプリケーションサーバにおいてネイティブアプリケーションを検索したユーザに、CMSにより開発したウェブアプリケーションを利用してもらうことができないという問題がある。これに対して、CMSによって構築したウェブアプリケーションと同等の機能を有するネイティブアプリケーションを開発し、当該ネイティブアプリケーションをアプリケーションサーバにアップロードすることも考えられる。しかしながら、ネイティブアプリケーションを新規に開発するには、多大な開発工数が必要であった。本発明は、ネイティブアプリケーションを容易に生成することができるアプリケーション生成装置、アプリケーション生成システム及びアプリケーション生成方法を提供することを目的とする。」(段落【0002】、【0004】～【0007】)と記載されており、同記載からすると、引用発明は、CMSによって構築されるウェブアプリケーションは、アプリケーションサーバから検索することができないため、アプリケーションサーバにおいてネイティブアプリケーションを検索したユーザに、CMSにより開発したウェブアプリケーションを利用してもらうことができないこと及びCMSによって構築したウェブアプリケーションと同等の機能を有するネイティブアプリケーションを新規に開発するには、多大な開発工数が必要となることを課題とし、同課題を解決するためのネイティブアプリケーションを生成する装置であることが認められる。

引用発明は、上記課題を解決するために、前記(1)アで認定したとおり、既存のウェブアプリケーションのロケーションを示すアドレスや所望の背景画像を示すアドレス等の情報を入力するだけで、当該ウェブアプリケーションの表示態様を変更して、同ウェブアプリケーションが表示する情報を表示するネイティブアプリケーシ

ョンを生成できるようにしたものと認められる。

イ 被告は、携帯通信端末の動きに伴う動作を行うネイティブアプリケーションを生成すること、特に、Phone Gapに係る技術が周知であると主張する。

(ア) 前記アのとおり、引用発明は、アプリケーションサーバにおいて検索できるネイティブアプリケーションを簡単に生成することを課題として、同課題を、既存のウェブアプリケーションのアドレス等の情報を入力するだけで、同ウェブアプリケーションが表示する情報を表示できるネイティブアプリケーションを生成することができるようにすることによって解決したものであるから、ブログ等の携帯通信端末の動きに伴う動作を行わないウェブアプリケーションの表示内容を表示するネイティブアプリケーションを生成しようとする場合、生成しようとするネイティブアプリケーションを携帯通信端末の動きに伴う動作を行うようにする必要はなく、したがって、設定ファイルを設定するパラメータを「携帯通信端末に固有のネイティブ機能を実行するためのパラメータ」とする必要はない。もっとも、引用文献1の段落【0024】には、ブログ等と並んで「ゲームサイト」が掲げられており、ゲームにおいては、加速度センサにより横画面と縦画面が切り替わらないように制御する必要がある場合が考えられる（引用文献5参照）が、ウェブアプリケーションとして提供されるゲームは、①常に携帯通信端末の表示画面を固定する必要があるとはいえないこと、②加速度センサにより、携帯通信端末の姿勢に対応した画面回転表示を制御する機能は携帯通信端末側に備わっており、端末側の操作によって、表示画面を固定することができ、そのような操作は一般的に行われていること、③引用文献1の段落【0024】の「ゲームサイト」は、携帯通信端末の表示画面を固定する必要のないブログ、ファンサイト、ショッピングサイトと並んで記載されており、また、引用文献1には、加速度センサについて何らの記載もないことからすると、当業者は、上記の「ゲームサイト」の記載から、パラメータを「携帯通信端末に固有のネイティブ機能を実行するためのパラメータ」とすることの必要性を認識するとまではいえないというべきである。

また、引用発明によって生成されるネイティブアプリケーションは、HTMLやJavaScriptで記述されるウェブページを表示できるから、引用発明により、乙4に記載されたHTML5 APIのGeolocationを用いて携帯通信端末の動きに伴う動作を行うウェブアプリケーションの表示内容を表示するネイティブアプリケーションを生成しようとする場合も、生成されるネイティブアプリケーションは、設定情報に含まれているウェブアプリケーションのアドレスに基づいて、同ウェブアプリケーションに対応するウェブページを取得し、取得したウェブページのHTMLやJavaScriptの記述に基づいて、同ウェブアプリケーションの内容を表示でき、したがって、ネイティブアプリケーションの生成に際して、設定ファイルを設定するパラメータを「携帯通信端末に固有のネイティブ機能を実行させるためのパラメータ」とする必要はない。

さらに、被告主張周知技術に係る各種文献にも、引用発明の上記の構成の技術において、「携帯通信端末に固有のネイティブ機能を実行させるためのパラメータ」に応じて設定ファイルを設定することの必要性等については何ら記載されていない（甲2～5，7，8，乙1～3）。

(イ) 前記アのとおり、引用発明は、簡易にネイティブアプリケーションを生成することを課題として、既存のウェブアプリケーションのアドレス等の情報を入力するだけで、当該ウェブアプリケーションが表示する情報を表示するネイティブアプリケーションを生成できるようにしたのであり、具体的には、前記(1)アのとおり、入力しようとするウェブアプリケーションのロケーションを示すアドレス及び表示態様に基づいて、テンプレートアプリケーション111に含まれる設定情報の内容を書き換えるだけで目的とするウェブアプリケーションの表示する情報を表示できるネイティブアプリケーションを生成でき、テンプレートアプリケーション111に含まれるプログラムファイル113については、新たにソースコードを書く必要はないところ、証拠（甲3，5，7，乙1～3）によると、PhoneGapによってネイティブアプリケーションを生成するためには、HTMLやJavaS

c r i p t 等を用いてソースコード（プログラム）を書くなどする必要があるものと認められるから，引用発明に，上記のように，新たにソースコードを書くなどの行為が要求される P h o n e G a p に係る技術を適用することには阻害事由があるというべきである。

被告は，① P h o n e G a p では， P h o n e G a p のプラグインの仕組みを使って， G P S など端末のネイティブ部分にアクセス可能であり，端末の G P S 機能にアクセスすることで， G P S で取得した端末の現在位置が中心となるように地図を表示することが可能となるから，引用発明において地図表示アプリケーションを生成する際に， P h o n e G a p のフレームワークを採用することで，ネイティブ機能である G P S 機能が利用可能となり，携帯通信端末の動きに伴う動作を設定可能となり，また，② A n d r o i d M a n i f e s t . x m l に「 a n d r o i d : s c r e e n O r i e n t a t i o n = " l a n d s c a p e " 」の1行を追加したり（ A n d r o i d の場合），「 L a n d s c a p e R i g h t 」又は「 L a n d s c a p e L e f t 」のアイコンを選択すること（ i O S の場合）で，スマートフォンの画面を横画面に固定可能となり，縦画面に固定する設定を施す場合， A n d r o i d M a n i f e s t . x m l に「 a n d r o i d : s c r e e n O r i e n t a t i o n = " p o r t r a i t " 」の1行を追加したり（ A n d r o i d の場合），「 p o r t r a i t 」のアイコンを選択すること（ i O S の場合）で，スマートフォンの画面を縦画面に固定可能となるから，引用発明において，アプリケーションを生成する際に， P h o n e G a p のフレームワークを利用することで，「携帯通信端末に固有のネイティブ機能を実行させるためのパラメータ」に応じて，携帯通信端末において実行される「アプリケーションの，携帯通信端末の動きに伴う動作」を規定する設定ファイルを備えることとなると主張する。

しかし，上記のとおり，引用発明に P h o n e G a p の技術を適用することの動機付けはないから，被告の上記主張は，その前提を欠くものであって，理由がない。

(ウ) 以上からすると，引用発明に，被告主張周知技術を適用することの動機

付けは認められないというべきである。

ウ 以上のとおり，引用発明に被告主張周知技術を適用することの動機付けはないから，引用発明に被告主張周知技術を適用して，相違点1の構成について，本件補正発明の構成とすることは容易に想到することはできず，したがって，本件補正発明は，引用発明及び被告主張周知技術に基づいて容易に発明することができたということとはできない。

(3) したがって，取消事由1は理由がある。

第6 結論

よって，主文のとおり判決する。

知的財産高等裁判所第2部

裁判長裁判官

森 義 之

裁判官

佐 野 信

裁判官

熊 谷 大 輔