

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第5534801号  
(P5534801)

(45) 発行日 平成26年7月2日 (2014.7.2)

(24) 登録日 平成26年5月9日 (2014.5.9)

(51) Int. Cl.

F I

G 0 6 F 9/445 (2006.01)

G 0 6 F 9/06 6 5 0 B

H 0 4 W 88/02 (2009.01)

G 0 6 F 9/06 6 4 0 A

H 0 4 W 88/02 1 6 0

請求項の数 3 (全 12 頁)

(21) 出願番号 特願2009-292828 (P2009-292828)  
 (22) 出願日 平成21年12月24日 (2009.12.24)  
 (65) 公開番号 特開2011-134095 (P2011-134095A)  
 (43) 公開日 平成23年7月7日 (2011.7.7)  
 審査請求日 平成24年11月29日 (2012.11.29)

(73) 特許権者 000001122  
 株式会社日立国際電気  
 東京都千代田区外神田四丁目14番1号  
 (74) 代理人 110000039  
 特許業務法人アイ・ピー・ウィン  
 (72) 発明者 金橋 祐輔  
 東京都羽村市神明台二丁目1番1号 株式  
 会社日立国際電気内  
 (72) 発明者 鈴木 道奉  
 東京都羽村市神明台二丁目1番1号 株式  
 会社日立国際電気内

審査官 坂庭 剛史

最終頁に続く

(54) 【発明の名称】 ソフトウェア無線装置

(57) 【特許請求の範囲】

【請求項1】

実行しうる複数の種類のオペレーティングシステムのいずれかと、  
少なくとも1つが実行されて無線通信処理を全体として実現する複数のソフトウェアコン  
ポーネントと、

前記オペレーティングシステム上において実行され、前記複数のソフトウェア内の少なく  
とも1つが、前記オペレーティングシステムの種類の種類と独立に実行され得るように  
するミドルウェアと

を実行する無線装置であって、

前記ミドルウェアは、

第1～第3のインタフェースコンポーネントを含み、前記複数のソフトウェアの前記オ  
ペレーティングシステムの種類の種類にかかわらず実行のために用いられる複数のイン  
タフェースコンポーネント

を備え、

前記第1～第3のインタフェースコンポーネント以外の前記複数のインタフェースコン  
ポーネントそれぞれは、前記無線通信処理に共通して使用される共通インタフェースが初  
期化され、前記共通インタフェースの共通インタフェース名とIPアドレスとが前記第2  
のインタフェースコンポーネントに登録された後に、それぞれのインタフェース名とIP  
アドレスとを、前記第2のインタフェースコンポーネントに登録し、

前記第1のインタフェースコンポーネントは、

10

20

前記無線装置が起動されるときに、前記共通インタフェースを起動して初期化し、  
前記初期化された共通インタフェースの共通インタフェース名とＩＰアドレスとを、前記複数のインタフェースコンポーネントの内の第２のインタフェースコンポーネントに登録し、

前記第２のインタフェースコンポーネントに登録された前記第１～第３のインタフェースコンポーネント以外の前記複数のインタフェースコンポーネントそれぞれのＩＰアドレスを取得し、

前記取得したＩＰアドレスを、前記第３のインタフェースコンポーネントに設定し、  
前記第３のインタフェースコンポーネントは、前記設定された前記第１～第３のインタフェースコンポーネント以外の前記複数のインタフェースコンポーネントそれぞれのＩＰアドレスに基づいて、これらのインタフェースコンポーネントに対応する前記ソフトウェアコンポーネントをダウンロードする

無線装置。

【請求項２】

実行しうる複数の種類のオペレーティングシステムのいずれかと、  
少なくとも１つが実行されて無線通信処理を全体として実現する複数のソフトウェアコンポーネントと、

前記オペレーティングシステム上において実行され、前記複数のソフトウェア内の少なくとも１つが、前記オペレーティングシステムの種類の種類と独立に実行され得るようにするミドルウェアと

を実行する無線装置であって、

前記無線装置は、

前記ソフトウェアコンポーネントを実行するＤＳＰと、このＤＳＰのレジスタとを含む複数のハードウェアデバイス

を含み、

前記ミドルウェアは、

第１，第４～第６のインタフェースコンポーネントを含み、前記複数のソフトウェアの前記オペレーティングシステムの種類の種類にかかわらず実行のために用いられる複数のインタフェースコンポーネント

を備え、

前記第１のインタフェースコンポーネントは、

前記第４のインタフェースコンポーネントに予め定められた前記ハードウェアデバイスに対応する前記ソフトウェアコンポーネントを指定して、前記指定したソフトウェアコンポーネントの起動を指示し、

前記第４のインタフェースコンポーネントは、前記第５のインタフェースコンポーネントに定義されたメソッドをコールして、前記指定されたソフトウェアコンポーネントを起動し、

前記第５のインタフェースコンポーネントは、

前記メソッドのコールに応じて、プロファイルを解析し、前記ＤＳＰのレジスタにおける書き込み領域を前記第６のインタフェースコンポーネントに指示し、

前記指示した書き込み領域へ前記ソフトウェアコンポーネントが書き込まれると、前記ＤＳＰのリセットを、前記第６のインタフェースコンポーネントに指示し、

前記第６のインタフェースコンポーネントは、

前記指定されたソフトウェアコンポーネントの書き込み領域の指定に応じて、前記指定されたＤＳＰのレジスタにおける書き込み領域へ前記ソフトウェアコンポーネントを書き込み、

前記ＤＳＰのリセットの指示に応じて、前記ＤＳＰを、前記ソフトウェアコンポーネントの実行のためにリセットする

無線装置。

【請求項３】

実行しうる複数の種類のオペレーティングシステムのいずれかと、  
少なくとも1つが実行されて無線通信処理を全体として実現する複数のソフトウェアコンポーネントと、  
前記オペレーティングシステム上において実行され、前記複数のソフトウェア内の少なくとも1つが、前記オペレーティングシステムの種類の種類と独立に実行され得るようにするミドルウェアと  
を実行する無線装置であって、  
前記無線装置は、  
前記ソフトウェアコンポーネントを実行するDSPと、このDSPのレジスタとを含む複数のハードウェアデバイス  
を含み、  
前記ミドルウェアは、  
第1～第6のインタフェースコンポーネントを含み、前記複数のソフトウェアの前記オペレーティングシステムの種類の種類にかかわらず実行のために用いられる複数のインタフェースコンポーネント  
を備え、  
前記第1～第3のインタフェースコンポーネント以外の前記複数のインタフェースコンポーネントそれぞれは、前記無線通信処理に共通して使用される共通インタフェースが初期化され、前記共通インタフェースの共通インタフェース名とIPアドレスとが前記第2のインタフェースコンポーネントに登録された後に、それぞれのインタフェース名とIP  
アドレスとを、前記第2のインタフェースコンポーネントに登録し、  
前記第1のインタフェースコンポーネントは、  
前記無線装置が起動されるときに、前記共通インタフェースを起動して初期化し、  
前記初期化された共通インタフェースの共通インタフェース名とIPアドレスとを、前記複数のインタフェースコンポーネントの内の第2のインタフェースコンポーネントに登録し、  
前記第2のインタフェースコンポーネントに登録された前記第1～第3のインタフェースコンポーネント以外の前記複数のインタフェースコンポーネントそれぞれのIPアドレスを取得し、  
前記取得したIPアドレスを、前記第3のインタフェースコンポーネントに設定し、  
前記第4のインタフェースコンポーネントに予め定められた前記ハードウェアデバイスに対応する前記ソフトウェアコンポーネントを指定して、前記指定したソフトウェアコンポーネントの起動を指示し、  
前記第4のインタフェースコンポーネントは、前記第5のインタフェースコンポーネントに定義されたメソッドをコールして、前記指定されたソフトウェアコンポーネントを起動し、  
前記第3のインタフェースコンポーネントは、前記設定された前記第1～第3のインタフェースコンポーネント以外の前記複数のインタフェースコンポーネントそれぞれのIPアドレスに基づいて、これらのインタフェースコンポーネントに対応する前記ソフトウェアコンポーネントをダウンロードし、  
前記第4のインタフェースコンポーネントは、前記第5のインタフェースコンポーネントに定義されたメソッドをコールして、前記指定されたソフトウェアコンポーネントを起動し、  
前記第5のインタフェースコンポーネントは、  
前記メソッドのコールに応じて、プロファイルを解析し、前記DSPのレジスタにおける書き込み領域を前記第6のインタフェースコンポーネントに指示し、  
前記指示した書き込み領域へ前記ソフトウェアコンポーネントが書き込まれると、前記DSPのリセットを、前記第6のインタフェースコンポーネントに指示し、  
前記第6のインタフェースコンポーネントは、  
前記指定されたソフトウェアコンポーネントの書き込み領域の指定に応じて、前記指定

10

20

30

40

50

されたDSPのレジスタにおける書き込み領域へ前記ソフトウェアコンポーネントを書き込み、

前記DSPのリセットの指示に応じて、前記DSPを、前記ソフトウェアコンポーネントの実行のためにリセットする

無線装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、ソフトウェア無線装置に関する。

10

【背景技術】

【0002】

例えば、引用文献1は、複数のモジュールから成るドメイン内で動作するソフトウェアプログラムの動作状況を管理するドメイン管理インタフェースを提供する。

【先行技術文献】

【特許文献】

【0003】

【特許文献1】特開2007-148844号公報

【発明の概要】

【発明が解決しようとする課題】

20

【0004】

本発明は、上述のような背景からなされたものであり、OSの違いを考慮することなく複数のソフトウェアを動作させることができるインタフェースを備え、低負荷で動作するプラットフォームを備えるよう改良された無線装置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】

【0005】

上記目的を達成するために、本発明に係る無線装置(1)は、オペレーティングシステム(208)と、複数のソフトウェアコンポーネント(200, 202)と、前記オペレーティングシステムと前記複数のソフトウェアコンポーネントとの間に備えられたミドルウェア(30)とを有し、前記複数のソフトウェアコンポーネントの1つ以上を実行して、全体として無線通信処理を行う無線装置であって、前記ミドルウェアに含まれ、この無線装置が起動されるときに、予め定められた条件に基づいて、前記複数のソフトウェアコンポーネントの1つ以上を起動させる第1の起動手段(308)と、前記ミドルウェアに含まれ、前記複数のソフトウェアコンポーネントのうち、外部から指定されたソフトウェアコンポーネントを起動させる第2の起動手段(310)とを有する。

30

なお、ここで付された符号は、本願発明の理解を助けることを意図するものであり、本願発明の技術的範囲を限定することを意図するものではない。

【発明の効果】

【0006】

本発明に係る無線装置によれば、OSの違いを考慮することなく複数のソフトウェアを動作させることができるインタフェースを備え、低負荷で動作するプラットフォームを実現することができる。

40

【図面の簡単な説明】

【0007】

【図1】(A)は、本発明に係るソフトウェア無線装置の構成を例示する図であり、(B)は、(A)に示したソフトウェア無線装置のプラットフォームレイヤの構成を例示する図である。

【図2】図1(B)に示したプラットフォームレイヤにおいて、ソフトウェアを実行させるサービスインタフェースの構成を例示する図である。

【図3】(A)は、図2に示したAutoFactoryインタフェースのクラス図を例

50

示する図であり、(B)は、AutoFactoryインタフェースに定義されるメソッド名または属性名、および、これらの説明を例示する図である。

【図4】(A)は、図2に示したMicroManagerインタフェースのクラス図を例示する図であり、(B)は、MicroManagerインタフェースに定義されるメソッド名または属性名、および、これらの説明を例示する図である。

【図5】図1(A)に示したソフトウェア無線装置の起動処理を例示するシーケンス図である。

【図6】図1(A)に示したソフトウェア無線装置におけるソフトウェアの交換処理を例示するシーケンス図である。

【発明を実施するための形態】

10

【0008】

[本発明の背景]

本発明の理解を助けるために、まず、本発明がなされるに至った背景を説明する。

例えば、無線装置は、ソフトウェア実行環境を備えた複数の演算処理装置と、CORBA(Common Object Request Broker Architecture)対応の複数のソフトウェアとを有し、無線通信処理の内容に応じて、複数のソフトウェアコンポーネントの中から必要なソフトウェアを、演算処理装置上の適切なソフトウェア実行環境に配置して実行するように構成される。

また、上述の無線装置には、複数のハードウェアデバイスの追加、削除などの操作を動的に管理するために、SCA(Software Communication Architecture)という標準化技術を利用するプラットフォームが構築される。

20

【0009】

例えば、SCAを利用するプラットフォームは、登録または削除されるハードウェアデバイスを動的に管理したり、ハードウェアデバイスの登録情報に基づいて、複数のソフトウェアから、無線通信処理に必要なソフトウェアを選択し、動的に配置したりするミドルウェアを備える。

上述のミドルウェアにより、複数のハードウェアデバイスや、複数のソフトウェアコンポーネントを動的に管理することができる一方で、このミドルウェアを動作させるためには、高性能のプロセッサを必要とする場合がある。

したがって、ハードウェアデバイスの追加などを行わない携帯電話などのプロセッサの性能では、上述のミドルウェアを備えるプラットフォームを動作させる負荷が高くなる場合がある。

30

【0010】

また、上述の複数のソフトウェアコンポーネントは、同一のOS(Operating System)上でしか動作しないので、異なるOS上で、同一のソフトウェアを動作させることができない。

したがって、OSの種類ごとに、同一の動作を行うソフトウェアを実装する必要があり、非効率的になる場合がある。

本発明に係る無線装置は、このような背景からなされたものであり、本発明に係る無線装置のプラットフォームは、OSの違いを考慮することなく複数のソフトウェアを動作させることができるインタフェースを備え、SCAを利用するプラットフォームよりも低負荷で動作するよう工夫されている。

40

【0011】

[無線装置1]

図1(A)は、本発明に係る無線装置1の構成を例示する図である。

図1(A)に示すように、無線装置1は、RF部102と、RF部102に接続されたアンテナ100と、CompactPCI(Peripheral Component Interconnect)などのバス110を介して接続されたモデム部12、コーデック部14および制御部16から構成される。

また、図1(A)に示すように、モデム部12は、CPU104およびメモリ106を

50

含み、コーデック部 1 4 および制御部 1 6 は、CPU 1 0 4、メモリ 1 0 6 およびネットワーク接続を行うための LAN インタフェース ( I F ) 1 0 8 を含む。

なお、これらの構成部分においては、CPU 1 0 4 とともに、あるいは、CPU 1 0 4 の代わりに、DSP (Digital Signal Processor) が用いられてもよい。

無線装置 1 は、制御部 1 6 により、モデム部 1 2 およびコーデック部 1 4 を制御して、必要な処理を行い、RF 部 1 0 2 およびアンテナ 1 0 0 を介して、予め定められた無線方式の信号を送受信して、無線通信処理を行う。

なお、以下の各図において、実質的に同じ構成部分には、同じ符号が付される。

#### 【 0 0 1 2 】

[ プラットフォームレイヤ 2 0 ]

10

図 1 ( B ) は、無線装置 1 ( 図 1 ( A ) ) のプラットフォームレイヤ 2 0 の構成を例示する図である。

図 1 ( B ) に示すように、無線装置 1 が備えるプラットフォームレイヤ 2 0 は、DSP ソフトウェア 2 0 0、GPP (General Purpose Processor) ソフトウェア 2 0 2、および、入出力ライブラリ 2 0 4、DSP のハードウェア 2 0 6、ミドルウェア 3 0、OS 2 0 8、および、GPP のハードウェア 2 1 0 からなるプラットフォームから構成される。

なお、以下、DSP ソフトウェア 2 0 0 および GPP ソフトウェア 2 0 2 を区別せず、単に「ソフトウェア」と記述することがある。

#### 【 0 0 1 3 】

なお、以下、本実施形態におけるプラットフォームレイヤ 2 0 のプラットフォームは、下記のポリシー ( 1 ) ~ ( 5 ) に従って実現される場合を具体例とする。

20

( 1 ) 1 つの GPP ボード上で実現する。

( 2 ) GPP ボードには、DSP や FPG A (Field Programmable Gate Array) を搭載することができる。

( 3 ) ソフトウェアは、予め定められた実行環境で実行される。つまり、複数の実行環境への動的なソフトウェア配置は行わない。

( 4 ) DSP や FPG A と通信するソフトウェアが常駐しているか否かに関わらず、プラットフォームのソフトウェア実行環境は利用可能である。

( 5 ) セキュリティプロキシを備えない。

#### 【 0 0 1 4 】

30

DSP ソフトウェア 2 0 0 は、無線通信処理を実現する DSP 用のソフトウェアコンポーネントである。

GPP ソフトウェア 2 0 2 は、無線通信処理を実現する GPP 用のソフトウェアコンポーネントである。

入出力ライブラリ 2 0 4 は、DSP ソフトウェア 2 0 0 と DSP のハードウェア 2 0 6 とを仲介するインタフェースであり、このインタフェースに定義されたメソッドがコールされることにより、双方に対するデータの入出力を実現する。

また、入出力ライブラリ 2 0 4 は、通信バスを介して、ミドルウェア 3 0 と通信を行い、各種処理を行う。

#### 【 0 0 1 5 】

40

通信バスは、例えば、バス 1 1 0 ( 図 1 ( A ) ) 内のソフトウェアバスである ORB (Object Request Broker) である。

ミドルウェア 3 0 は、OS 2 0 8 上で動作し、GPP ソフトウェア 2 0 2 と OS 2 0 8 とを仲介する複数のインタフェース ( 図 2 を参照して後述 ) から構成される。

GPP ソフトウェア 2 0 2 は、これらの複数のインタフェースに定義されたメソッドをコールすることにより、OS 2 0 8 の違いを考慮することなく、OS 2 0 8 上で、GPP のハードウェア 2 1 0 の資源を具体的に利用して実行するよう処理される。

また、ミドルウェア 3 0 は、通信バスを介して、入出力ライブラリ 2 0 4 と通信を行い、各種処理を行う。

#### 【 0 0 1 6 】

50

### [ サービスインタフェース 3 ]

図 2 は、プラットフォームレイヤ 20 ( 図 1 ( B ) ) において、OS 208 の違いを考慮することなく、ソフトウェアを実行させるサービスインタフェース 3 の構成を例示する図である。

図 2 に示すように、サービスインタフェース 3 は、1 つの G P P ボード上に搭載された DSP 用の入出力ライブラリ 204、および、CPU 用のミドルウェア 30 から構成される。

また、ミドルウェア 30 は、H C I ( Host Controller Interface ) 300、N a m i n g S e r v i c e 302、L o g 304、G P P \_ D e v i c e 306、A u t o F a c t o r y 308、M i c r o M a n a g e r 310、および、D o w n L o a d e r 312 などの CORBA 対応の複数のインタフェースを備える。

10

#### 【 0017 】

H C I 300 は、無線装置 1 ( 図 1 ( B ) ) の起動時に、必要なソフトウェアを起動させるための処理を行うインタフェースである。

例えば、H C I 300 は、N a m i n g S e r v i c e 302 に起動させるソフトウェアの IP アドレスを要求して取得し、D o w n L o a d e r 312 に対して、取得した IP アドレスを設定する。

また、H C I 300 は、予め定められたハードウェアデバイスに対応するソフトウェアを起動させるための処理を行う。

N a m i n g S e r v i c e 302 は、各種インタフェース名と、インタフェースの IP アドレスを登録し、要求に応じて、登録されたアドレスを要求元に対して出力するインタフェースである。

20

#### 【 0018 】

G P P \_ D e v i c e 306 は、DSP、シリアルデバイス、オーディオデバイスなどの CORBA 非対応のハードウェアデバイスにアクセス可能とする論理デバイスインタフェースである。

A u t o F a c t o r y 308 は、無線装置 1 ( 図 1 ( A ) ) の次回起動時に、今回最後に実行されているソフトウェアを自動的に起動させるよう設定するインタフェースである ( 図 3 を参照して後述 ) 。

M i c r o M a n a g e r 310 は、ソフトウェアの起動および終了を指示し、ソフトウェアのリストを管理するインタフェースである ( 図 4 を参照して後述 ) 。

30

D o w n L o a d e r 312 は、H C I 300 から設定された IP アドレスに基づいて、IP アドレスに対応するソフトウェアをダウンロードするインタフェースである。

#### 【 0019 】

### [ A u t o F a c t o r y 308 ]

図 3 ( A ) は、サービスインタフェース 3 ( 図 2 ) に含まれる A u t o F a c t o r y インタフェース 308 のクラス図を例示する図であり、図 3 ( B ) は、A u t o F a c t o r y インタフェース 308 に定義されるメソッド名または属性名、および、これらの説明を例示する図である。

図 3 ( A ) に示すように、A u t o F a c t o r y インタフェース 308 は、A p p l i c a t i o n F a c t o r y インタフェースを継承する。

40

A p p l i c a t i o n F a c t o r y インタフェースは、S C A を利用するプラットフォームが備えるミドルウェアであり、例えば、無線通信処理の内容に応じて、必要なソフトウェアを選択して、選択したソフトウェアを、複数の実行環境のいずれかに配置したり、起動したりする。

#### 【 0020 】

A u t o F a c t o r y インタフェース 308 は、上述のポリシー ( 3 ) に従って、複数の実行環境への動的なソフトウェア配置を行わず、無線装置 1 ( 図 1 ( A ) ) の次回起動時に、今回最後に実行されているソフトウェアを自動的に起動させるよう設定する。

図 3 ( B ) に示すように、A u t o F a c t o r y インタフェース 308 には、例えば

50

、ソフトウェアの自動起動の設定がオンであるかオフであることを示す属性 `AutoOn__OFF` を定義する。

また、属性 `AutoOn__OFF` が、オンを示す場合は、次回起動時に、今回最後に実行されているソフトウェアを自動的に起動させるよう設定する `Auto__create__at__start` メソッドを定義する。

#### 【0021】

[ `MicroManager310` ]

図4(A)は、サービスインタフェース3(図2)に含まれる `MicroManager` インタフェース310のクラス図を例示する図であり、図4(B)は、`MicroManager310` に定義されるメソッド名または属性名、および、これらの説明を例示する図である。

10

図4(A)に示すように、`MicroManager` インタフェース310は、`DomainManager` インタフェースと集約関係にある。

`DomainManager` インタフェースは、SCAを利用するプラットフォームが備えるミドルウェアであり、例えば、`GPP_Device306`(図2)などの各種インタフェースを管理し、管理されたインタフェースのメソッドのコールに応じて、ソフトウェアに対する処理や管理を行う。

#### 【0022】

`MicroManager` インタフェース310は、`DomainManager` インタフェースに定義された各種処理のうち、ソフトウェアの起動および終了を指示する処理と、ソフトウェアのリスト管理とを行う。

20

図4(B)に示すように、`MicroManager` インタフェース310には、例えば、指定したソフトウェアを起動させる `launchApplication` メソッドや、指定したソフトウェアを終了させる `cleanApplication` メソッドを定義する。

#### 【0023】

[ 無線装置1の第1の動作例 ]

図5は、無線装置1(図1(A))の起動処理を例示するシーケンス図である。

以下、図5を参照して、無線装置1の起動処理を、第1の動作例として説明する。

図5に示すように、ステップ400(S400)において、`HCI300`は、無線装置1において、無線通信処理の内容に関わらず、共通して使用される共通インタフェースを起動し、初期化する。

30

#### 【0024】

ステップ402(S402)において、`HCI300`(図2)は、S400において起動・初期化された共通インタフェース名と、IPアドレスとを、`NamingService302` に対して登録処理する。

ステップ404(S404)において、3つのインタフェース `MicroManager310`、`Log304` および `GPP_Device306` は、それぞれ起動処理を行い、インタフェース名と、IPアドレスとを、`NamingService302` に対して登録処理する。

40

ステップ406(S406)において、`Downloader312` は、起動処理を行い、インタフェース名と、IPアドレスとを、`NamingService302` に対して登録処理する。

#### 【0025】

ステップ408(S408)において、`HCI300`は、`NamingService302` に対して、`MicroManager310`、`Log304` および `GPP_Device306` のIPアドレスを要求して、取得する。

ステップ410(S410)において、`HCI300`は、S408において取得した各インタフェースのIPアドレスを、`Downloader312` に対して設定する。

ステップ412(S412)において、`Downloader312` は、S410にお

50



いてH C I 3 0 0 から設定されたI P アドレスに基づいて、I P アドレスに対応するソフトウェアをダウンロードする。

ステップ4 1 4 ( S 4 1 4 ) において、H C I 3 0 0 は、起動完了処理を行う。

【 0 0 2 6 】

[ 無線装置 1 の第 2 の動作例 ]

図 6 は、無線装置 1 ( 図 1 ( A ) ) におけるソフトウェアの交換処理を例示するシーケンス図である。

ソフトウェアの交換処理は、G P P \_ D e v i c e 3 0 6 ( 図 2 ) が別のハードウェアデバイスまたは最新のハードウェアデバイスにアクセス可能になるようにすることである。

以下、図 6 を参照して、無線装置 1 におけるソフトウェアの交換処理を、第 2 の動作例として説明する。

【 0 0 2 7 】

ステップ5 0 0 ( S 5 0 0 ) において、H C I 3 0 0 ( 図 2 ) は、M i c r o M a n a g e r 3 1 0 のl a u n c h A p p l i c a t i o n メソッドをコールして、予め定められたハードウェアデバイスに対応するソフトウェアを指定して、指定したソフトウェアの起動処理を指示する。

ステップ5 0 2 ( S 5 0 2 ) において、M i c r o M a n a g e r 3 1 0 は、A u t o F a c t o r y 3 0 8 のc r e a t e メソッドをコールして、S 5 0 0 において指定されたソフトウェアの起動処理を行う。

ステップ5 0 4 ( S 5 0 4 ) において、A u t o F a c t o r y 3 0 8 は、プロファイルを解析する。

【 0 0 2 8 】

ステップ5 0 6 ( S 5 0 6 ) において、A u t o F a c t o r y 3 0 8 は、G P P \_ D e v i c e 3 0 6 のメソッドをコールして、D S P レジスタのソフトウェアの書き込み領域の割り当て処理を指示する。

ステップ5 0 8 ( S 5 0 8 ) において、A u t o F a c t o r y 3 0 8 は、G P P \_ D e v i c e 3 0 6 のメソッドをコールして、ソフトウェアの書き込み処理を指示する。

ステップ5 1 0 ( S 5 1 0 ) において、G P P \_ D e v i c e 3 0 6 は、S 5 0 6 において割り当てたD S P レジスタの領域へ、S 5 0 8 において指示されたソフトウェアの書き込み処理を行う。

【 0 0 2 9 】

ステップ5 1 2 ( S 5 1 2 ) において、A u t o F a c t o r y 3 0 8 は、G P P \_ D e v i c e 3 0 6 のメソッドをコールして、D S P のリセット処理を指示する。

ステップ5 1 4 ( S 5 1 4 ) において、G P P \_ D e v i c e 3 0 6 は、S 5 1 2 におけるA u t o F a c t o r y 3 0 8 からの指示を受けて、D S P のリセット処理を行う。

ステップ5 1 6 ( S 5 1 6 ) において、H C I 3 0 0 は、ソフトウェア交換処理の完了通知を行う。

【 0 0 3 0 】

[ 無線装置 1 の特徴 ]

以下、無線装置 1 ( 図 1 ( A ) ) の特徴を説明する。

以上説明したように、無線装置 1 のプラットフォームは、ソフトウェアを自動的に起動するインタフェース、および、指定したソフトウェアの起動処理を指示するインタフェースなどから構成されるミドルウェアを備える。

このようなインタフェースにより、S C A を利用するプラットフォームが有するインタフェースのように、複数のソフトウェアから動的に適切なソフトウェアを選択して、配置する必要がないので、低負荷でプラットフォームを動作させることができる。

また、無線装置 1 のプラットフォームは、O S とソフトウェアの間に、このようなインタフェースから構成されるミドルウェアを備えることにより、O S の違いを考慮することなく、複数のソフトウェアを動作させることができる。

10

20

30

40

50

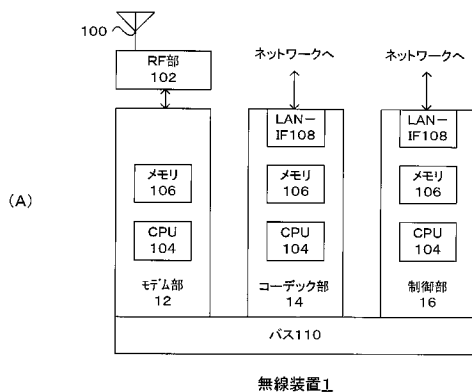
したがって、本発明に係る無線装置１によれば、ＯＳの違いを考慮することなく複数のソフトウェアを動作させることができるインタフェースを有し、ＳＣＡを利用するプラットフォームよりも低負荷で動作するプラットフォームを実現することができる。

【符号の説明】

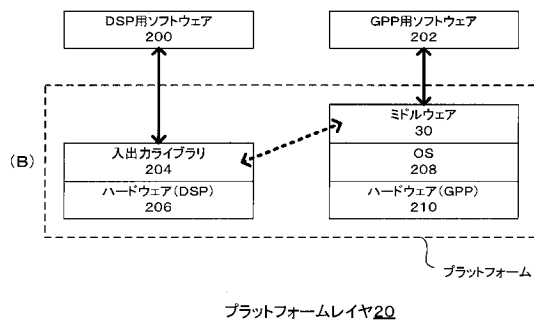
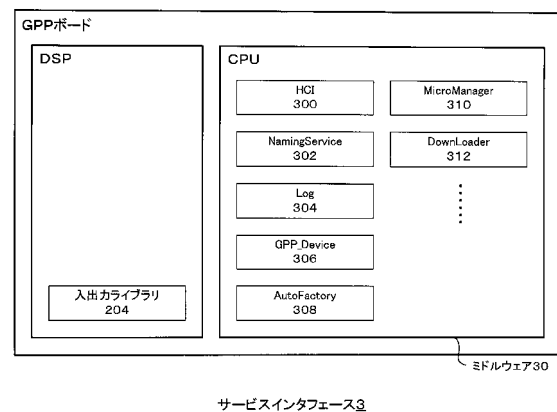
【００３１】

１・・・無線装置，１２・・・モデム部，１４・・・コーデック部，１６・・・制御部，１００・・・アンテナ，１０２・・・ＲＦ部，１０４・・・ＣＰＵ，１０６・・・メモリ，１０８・・・ＬＡＮ－ＩＦ，１１０・・・バス，２０・・・プラットフォームレイヤ，２００・・・ＤＳＰソフトウェア，２０２・・・ＧＰＰソフトウェア，２０４・・・入出力ライブラリ，２０６・・・ＤＳＰハードウェア，２０８・・・ＯＳ，２１０・・・Ｇ  
 １０  
 ＧＰＰハードウェア，３・・・サービスインタフェース，３０・・・ミドルウェア，３００・・・ＨＣＩ，３０２・・・ＮamingService，３０４・・・Log，３０６・・・ＧＰＰ\_Device，３０８・・・AutoFactory，３１０・・・MicroManager，３１２・・・Downloader

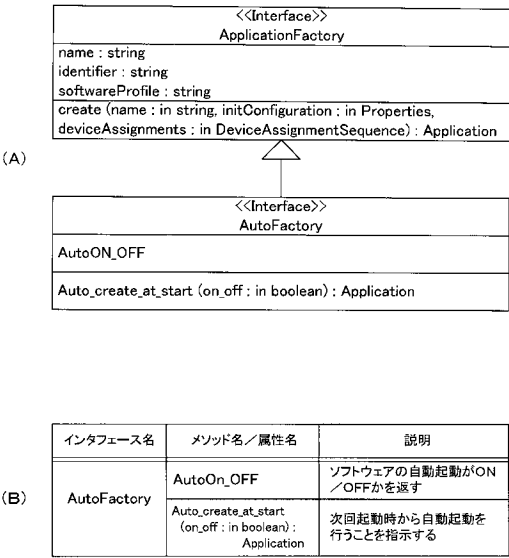
【図１】



【図２】



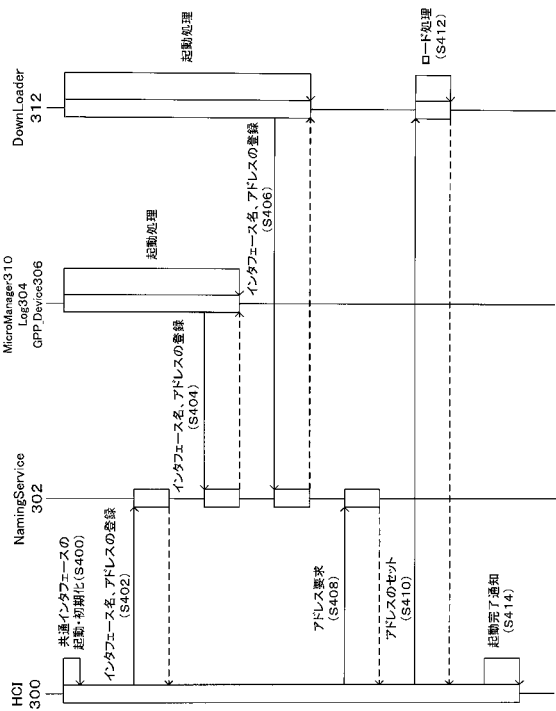
【図 3】



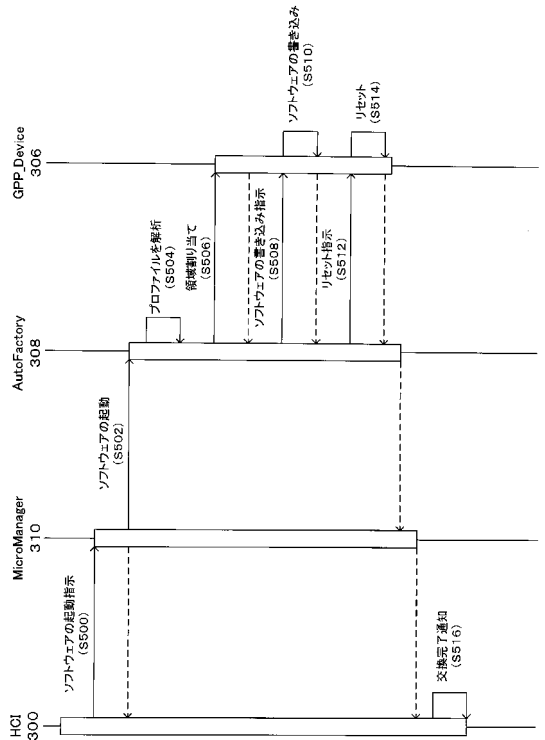
【図 4】



【図 5】



【図 6】



## フロントページの続き

(56)参考文献 特開2004-334679(JP,A)

特開2003-108382(JP,A)

特開2009-245428(JP,A)

特開2009-223732(JP,A)

特開2007-228461(JP,A)

特開2007-148844(JP,A)

特開2006-331307(JP,A)

特開2006-309533(JP,A)

特開2005-039557(JP,A)

特開2004-334735(JP,A)

特開平06-202861(JP,A)

白須光雄、関向賢一、金橋祐輔、中村 聡、其阿彌光行、藤井拓三、ソフトウェア無線機の製品化、電子情報通信学会技術研究報告、日本、社団法人電子情報通信学会、2006年 7月20日、Vol.106, No.188, pp.104~106, SR2006-28

金橋祐輔、白須光雄、関向賢一、中村 聡、其阿彌光行、藤井拓三、ラックマウント型ソフトウェア無線機の製品化、電子情報通信学会技術研究報告、日本、社団法人電子情報通信学会、2009年 7月22日、Vol.109, No.155, pp.149~150, SR2009-45

金橋祐輔、手嶋 功、中村 聡、其阿彌光行、藤井拓三、SCA(Software Communication Architecture)v2.2に基づいたソフトウェア無線機のソフトウェアアーキテクチャとデバイス関連インタフェース、電子情報通信学会技術研究報告、日本、社団法人電子情報通信学会、2005年 5月4日、Vol.105, No.36, pp.95~96, SR2005-14

(58)調査した分野(Int.Cl., DB名)

G06F 9/445

H04W 88/02