

(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.

G06F 9/445 (2006.01)  
H04W 88/02 (2009.01)

F 1

G06F 9/06 650B  
G06F 9/06 640A  
H04W 88/02 160

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・ワイン
(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のインター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ない実行のために用いられる複数のインターフェースコンポーネント

を備え、

前記第1～第3のインターフェースコンポーネント以外の前記複数のインター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ーフェースが初期化され、前記共通インターフェースの共通インターフェース名とIPアドレスとが前記第2のインターフェースコンポーネントに登録された後に、それぞれのインターフェース名とIPアドレスとを、前記第2のインターフェースコンポーネントに登録し、

前記第1のインター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、前記共通インター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ーフェースの共通インターフェース名とIPアドレスとを、前記複数のインターフェースコンポーネントの内の第2のインターフェースコンポーネントに登録し、

前記第2のインターフェースコンポーネントに登録された前記第1～第3のインターフェースコンポーネント以外の前記複数のインター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のインターフェースコンポーネントに設定し、前記第3のインター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のインターフェースコンポーネント以外の前記複数のインター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、これらのインター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のインター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ない実行のために用いられる複数のインターフェースコンポーネント

を備え、

前記第1のインターフェースコンポーネントは、

前記第4のインター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のインターフェースコンポーネントは、前記第5のインター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のインター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のインター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のインターフェースコンポーネントに指示し、

前記第6のインター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】

10

20

30

40

50

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

10

を含み、

前記ミドルウェアは、

第1～第6のインター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ない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のインターフェースコンポーネント以外の前記複数のインターフェースコン  
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ーフェースが初  
期化され、前記共通インターフェースの共通インターフェース名とIPアドレスとが前記第2  
のインターフェースコンポーネントに登録された後に、それぞれのインターフェース名とIP  
アドレスとを、前記第2のインターフェースコンポーネントに登録し、

20

前記第1のインター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、前記共通インター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ーフェースの共通インターフェース名とIPアドレスとを、前  
記複数のインターフェースコンポーネントの内の第2のインターフェースコンポーネントに登  
録し、

前記第2のインターフェースコンポーネントに登録された前記第1～第3のインターフェー  
スコンポーネント以外の前記複数のインター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、

30

前記取得したIPアドレスを、前記第3のインターフェースコンポーネントに設定し、

前記第4のインター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のインターフェースコンポーネントは、前記第5のインター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のインター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интер  
фейсコンポーネント以外の前記複数のインター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、これらのインター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、

40

前記第4のインターフェースコンポーネントは、前記第5のインター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のインター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のインター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のインターフェースコンポーネントに指示し、

前記第6のインター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

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ー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ーフェースを備え、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ーフェースを備え、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インターフェースのクラス図を例

50

示する図であり、(B)は、AutoFactoryインター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インターフェースのクラス図を例示する図であり、(B)は、MicroManagerインター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ーフェースを備え、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14および制御部16は、CPU104、メモリ106およびネットワーク接続を行うためのLANインターフェース(IF)108を含む。

なお、これらの構成部分においては、CPU104とともに、あるいは、CPU104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16により、モデム部12およびコーデック部14を制御して、必要な処理を行い、RF部102およびアンテナ100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。

#### 【0012】

##### [ プラットフォームレイヤ20 ]

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20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20は、DSPソフトウェア200、GPP(General Purpose Processor)ソフトウェア202、および、入出力ライブラリ204、DSPのハードウェア206、ミドルウェア30、OS208、および、GPPのハードウェア210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200およびGPPソフトウェア202を区別せず、単に「ソフトウェア」と記述することがある。

#### 【0013】

なお、以下、本実施形態におけるプラットフォームレイヤ20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FPGA(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FPGA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。

#### 【0014】

30

DSPソフトウェア200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202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204は、DSPソフトウェア200とDSPのハードウェア206とを仲介するインターフェースであり、このインター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204は、通信バスを介して、ミドルウェア30と通信を行い、各種処理を行う。

#### 【0015】

40

通信バスは、例えば、バス110(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30は、OS208上で動作し、GPPソフトウェア202とOS208とを仲介する複数のインターフェース(図2を参照して後述)から構成される。

GPPソフトウェア202は、これらの複数のインター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、OS208の違いを考慮することなく、OS208上で、GPPのハードウェア210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30は、通信バスを介して、入出力ライブラリ204と通信を行い、各種処理を行う。

#### 【0016】

50

[ サービスインターフェース 3 ]

図 2 は、プラットフォームレイヤ 20 ( 図 1 ( B ) )において、OS208 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ーフェース 3 の構成を例示する図である。

図 2 に示すように、サービスインターフェース 3 は、1 つの GPP 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、HCI ( Host Controller Interface ) 300 、 NamingService 302 、 Log 304 、 GPP\_Device 306 、 AutoFactory 308 、 MicroManager 310 、および、Downloader 312 などの CORBA 対応の複数のインターフェースを備える。  
10

【 0017 】

HCI 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ーフェースである。

例えば、HCI 300 は、NamingService 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、Downloader 312 に対して、取得した IP アドレスを設定する。

また、HCI 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。

NamingService 302 は、各種インターフェース名と、インターフェースの 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ーフェースである。  
20

【 0018 】

GPP\_Device 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ーフェースである。

AutoFactory 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ーフェースである ( 図 3 を参照して後述 ) 。

MicroManager 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ーフェースである ( 図 4 を参照して後述 ) 。  
30

Downloader 312 は、HCI 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ーフェースである。

【 0019 】

[ AutoFactory 308 ]

図 3 ( A ) は、サービスインターフェース 3 ( 図 2 ) に含まれる AutoFactory 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、AutoFactory 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、AutoFactory インタフェース 308 は、ApplicationFactory インタフェースを継承する。  
40

ApplicationFactory 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、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 】

AutoFactory 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、AutoFactory 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かを示す属性 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、サービスインターフェース3(図2)に含まれるMicroManagerインターフェース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インターフェース310は、DomainManagerインターフェースと集約関係にある。

DomainManagerインターフェースは、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)などの各種インターフェースを管理し、管理されたインター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インターフェース310は、DomainManagerインター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インターフェース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ーフェースを起動し、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ーフェース名と、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つのインターフェースMicroManager310、Log304およびGPP\_Device306は、それぞれ起動処理を行い、インターフェース名と、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、インターフェース名と、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ーフェースの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ーフェース、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ーフェースなどから構成されるミドルウェアを備える。

このようなインターフェースにより、 S C A を利用するプラットフォームが有するインターフェースのように、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、このようなインター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ーフェースを有し、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。

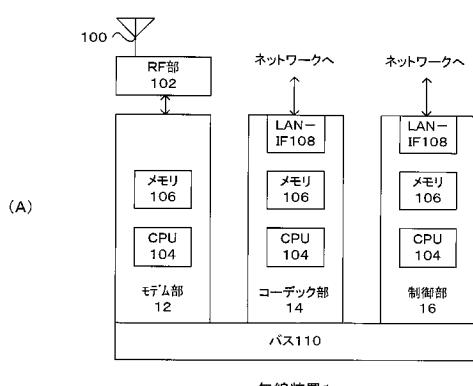
#### 【符号の説明】

##### 【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・・・GPPハードウェア，3・・・サービスインターフェース，30・・・ミドルウェア，300・・・HCI，302・・・NamingService，304・・・Log，306・・・GPP\_Device，308・・・AutoFactory，310・・・MicroManager，312・・・Downloader

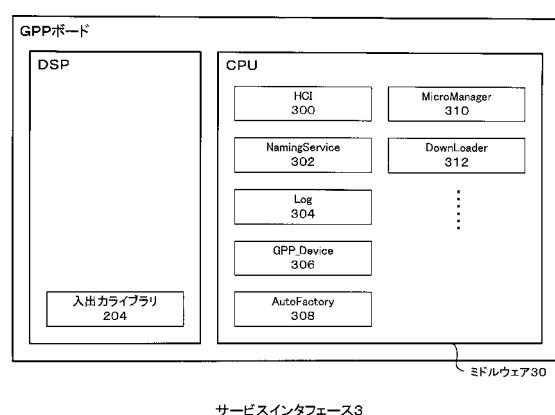
10

【図1】

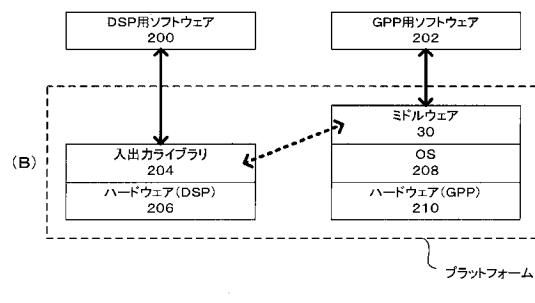


無線装置1

【図2】

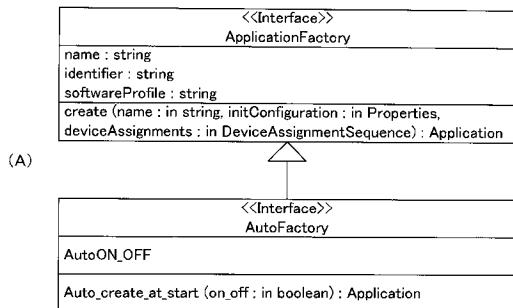


サービスインターフェース30

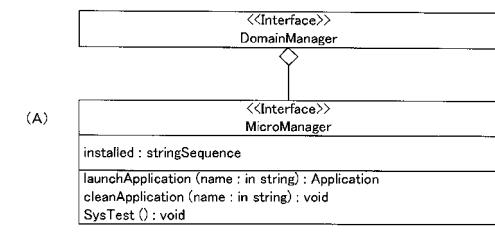


プラットフォームレイヤ20

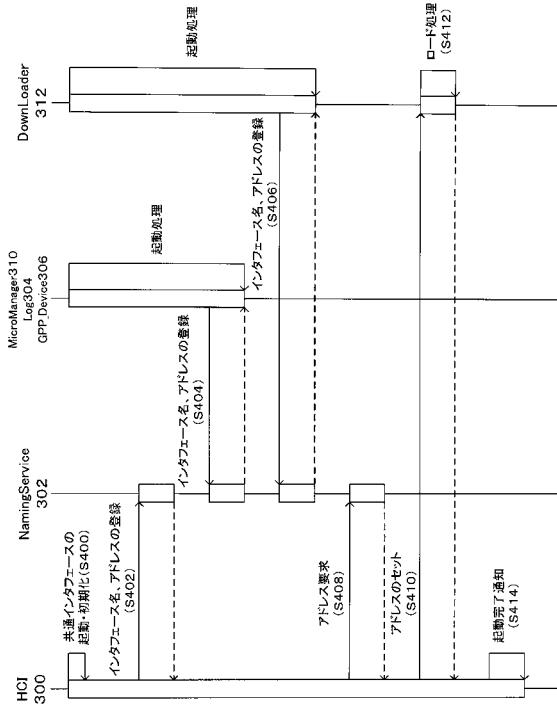
【図3】



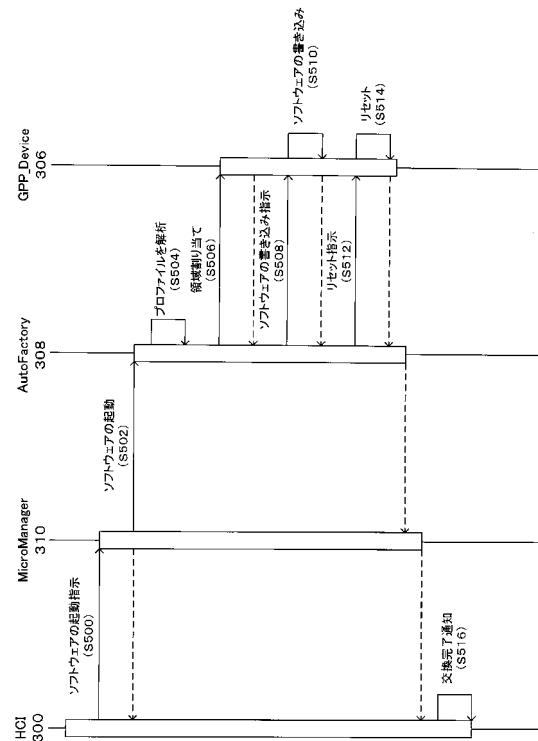
【図4】



【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 聰、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 聰、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 聰、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ーフェース、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)

G 06 F 9 / 445

H 04 W 88 / 02