

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第5669433号
(P5669433)

(45) 発行日 平成27年2月12日(2015.2.12)

(24) 登録日 平成26年12月26日(2014.12.26)

(51) Int.Cl.

G06F 9/44 (2006.01)

F 1

G06F 9/06 620K

請求項の数 11 (全 19 頁)

(21) 出願番号 特願2010-106665 (P2010-106665)
 (22) 出願日 平成22年5月6日 (2010.5.6)
 (65) 公開番号 特開2011-237883 (P2011-237883A)
 (43) 公開日 平成23年11月24日 (2011.11.24)
 審査請求日 平成25年3月5日 (2013.3.5)

(73) 特許権者 000001007
 キヤノン株式会社
 東京都大田区下丸子3丁目30番2号
 (74) 代理人 100076428
 弁理士 大塚 康徳
 (74) 代理人 100112508
 弁理士 高柳 司郎
 (74) 代理人 100115071
 弁理士 大塚 康弘
 (74) 代理人 100116894
 弁理士 木村 秀二
 (74) 代理人 100130409
 弁理士 下山 治
 (74) 代理人 100134175
 弁理士 永川 行光

最終頁に続く

(54) 【発明の名称】 部品結合装置及び部品結合方法

(57) 【特許請求の範囲】

【請求項 1】

アプリケーションを構成する部品と外部サービスとを結合する方法であって、
前記外部サービスのインターフェースの情報を取得する取得工程と、
 他の部品と結合された処理対象の部品が他の部品のサービスを利用するためのインターフェースに、前記取得工程で取得した情報が示す外部サービスのインターフェースが適合するか判断する判断工程と、

前記判断工程によって、前記処理対象の部品が前記他の部品を利用するためのインターフェースに、前記取得工程で取得した情報が示す外部サービスのインターフェースが適合するか判断された場合、前記処理対象の部品が前記外部サービスと結合されるように、前記処理対象の部品と前記他の部品との結合を変更する変更工程と、

前記外部サービスと結合された部品を実行する実行工程と、
 を備えることを特徴とする方法。

【請求項 2】

前記判断工程では、前記処理対象の部品が利用可能な部品を示す設定に基づいて選択された外部サービスのインターフェースが前記インターフェースに適合するか判断することを特徴とする請求項 1 記載の方法。

【請求項 3】

前記判断工程では、前記処理対象の部品が利用可能な外部サービスを外部装置に出力し、前記外部装置で指定された外部サービスのインターフェースが前記インターフェースに適合

するか判断する

ことを特徴とする請求項 1 記載の方法。

【請求項 4】

前記変更工程では、前記処理対象の部品と前記他の部品との結合の変更として、前記処理対象の部品と結合された他の部品と、該処理対象の部品と結合する候補の部品とが記述された定義を、変更する

ことを特徴とすることを特徴とする請求項 1 記載の方法。

【請求項 5】

前記変更工程では、前記処理対象の部品が前記外部サービスと結合され、前記他の部品が結合する候補の部品として記述されるように、前記処理対象の部品と前記他の部品との結合の定義を変更する

ことを特徴とする請求項 1 記載の方法。

【請求項 6】

前記変更工程では、前記処理対象の部品と前記他の部品との結合の変更として、サービスコンポーネントアーキテクチャー (S C A) 仕様に従って記述された、前記処理対象の部品と前記他の部品との結合の関係を示す結合定義を変更する

ことを特徴とする請求項 1 記載の方法。

【請求項 7】

前記判断工程では、前記処理対象の部品が前記他の部品のサービスを利用するためのインターフェースのクラスと前記外部サービスのクラスが対応するか、判断する

ことを特徴とする請求項 1 記載の方法。

【請求項 8】

前記判断工程では、優先使用部品として指定された前記外部サービスのインターフェースが、前記他の部品のサービスを利用するためのインターフェースに適合するか、判断する

ことを特徴とする請求項 1 記載の方法。

【請求項 9】

前記処理対象の部品が他の部品のサービスを利用するためのインターフェースは、 J a v a インタフェースクラスに従う

ことを特徴とする請求項 1 記載の方法。

【請求項 10】

アプリケーションを構成する部品と外部サービスとを結合する部品結合装置の制御をコンピュータに機能させるためのプログラムであって、

前記コンピュータに、

前記外部サービスのインターフェースの情報を取得する取得手段と、

他の部品と結合された処理対象の部品が他の部品のサービスを利用するためのインターフェースに、前記取得手段で取得した情報が示す外部サービスのインターフェースが適合するか判断する判断手段と、

前記判断手段によって、前記処理対象の部品が前記他の部品を利用するためのインターフェースに、前記取得手段で取得した情報が示す外部サービスのインターフェースが適合するか判断された場合、前記処理対象の部品が前記外部サービスと結合されるように、前記処理対象の部品と前記他の部品との結合を変更する変更手段と、

前記外部サービスと結合された部品を実行する実行手段と、

して機能させることを特徴とするプログラム。

【請求項 11】

アプリケーションを構成する部品と外部サービスとを結合する部品結合装置であって、前記外部サービスのインターフェースの情報を取得する取得手段と、

他の部品と結合された処理対象の部品が他の部品のサービスを利用するためのインターフェースに、前記取得手段で取得した情報が示す外部サービスのインターフェースが適合するか判断する判断手段と、

前記判断手段によって、前記処理対象の部品が前記他の部品を利用するためのインターフェース

10

20

30

40

50

エースに、前記取得手段で取得した情報が示す外部サービスのインターフェースが適合すると判断された場合、前記処理対象の部品が前記外部サービスと結合されるように、前記処理対象の部品と前記他の部品との結合を変更する変更手段と、

前記外部サービスと結合された部品を実行する実行手段と、
を備えることを特徴とする部品結合装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、ソフトウェア部品を結合してアプリケーションを構築する装置に関し、特に結合させるソフトウェア部品を置換又は変更する技術に関する。

10

【背景技術】

【0002】

従来、ソフトウェア部品（以下、単に部品と称す）又はコンポーネントを組み合わせてアプリケーションを構築する多様な技術が存在する。

【0003】

Open SCA Collaborationが仕様策定したサービスコンポーネントアーキテクチャー仕様（Service Component Architecture）はそのような技術の一つである。部品（SCA用語ではコンポーネント）のインターフェースを、サービス、リファレンス、プロパティに単純化し、異言語部品や多様な通信プロトコルを組み合わせし易くしたところに特徴がある。その組み合わせ仕様の詳細は非特許文献1に記載されている。

20

【0004】

SCA仕様では、部品の結合をXML言語で記述された定義ファイルを用いて行うよう既定されている。ある部品が別の部品を利用する場合、それらの部品は定義ファイルが定める方法によって結合されるが、利用する側の部品は利用される側の部品の機能を直接呼び出すようなコードを書く必要はない。

【0005】

従って、部品のインターフェースが一致していれば、アプリケーションの中で使用されている部品を別の部品に置き換える時にコードを書き直す必要はない。これはSCA技術を使うメリットである。

【0006】

30

一方、SCA以外の従来の技術として、部品を結合してアプリケーションを構築させる際に部品の置換を行ったり結合させるべき部品間の整合性を検証したりする技術に、以下のようなものが挙げられる。

【0007】

例えば、アプリケーション毎に、結合するモジュールの情報を管理・更新し、問題発生時や自動切替時に、モジュール情報から切替るべきモジュールのバージョンを決定し、そのアプリケーションが結合するモジュールバージョンの切り替えを行う。例えば、特許文献1参照。

【0008】

また、ソフトウェア内の各既開発プログラム部品間の組み合わせ情報および部品の属性情報に基づいて、開発済みのソフトウェアの構造を保持しつつ、当該ソフトウェアの一部をなす特定の既開発プログラム部品を新たな既開発プログラム部品に置換する。例えば、特許文献2参照。

40

【0009】

また、コンポーネント呼び出しを監視するモニタにより抽出されたコンポーネント呼び出し情報に基づいて、コンポーネント呼び出しとその対象であるコンポーネントとの整合性を検証する。例えば、特許文献3参照。

【先行技術文献】

【特許文献】

【0010】

50

【特許文献 1】特開 2008 - 108155 号公報

【特許文献 2】特開 2002 - 366356 号公報

【特許文献 2】特開 2004 - 362000 号公報

【非特許文献】

【0011】

【非特許文献 1】SCA Service Component Architecture Assembly Model Specification Version 1.00, March 15 2007

【発明の概要】

【発明が解決しようとする課題】

【0012】

部品の置き換えがし易いという SCA の特徴は好ましい。しかしながら、SCA 仕様では、アプリケーションを構築する際に、定義ファイルの内容を決定しておかなければならぬ。定義ファイルはアプリケーションの一部だからである。つまり、部品の置き換えを行う場合は、定義ファイルを変更せねばならない。それはアプリケーションの構築し直しを意味する。アプリケーションの実行時ではなく、開発時に部品の置き換えが可能ということである。

【0013】

部品の置き換えの柔軟性はできるだけ高めておくことが望ましい。たとえ部品のインターフェースが同一であったとしても、性能や使用リソース、スケーラビリティ、アベイラビリティなど、部品の特性としてはいくつも考えられ、実現したい特性に応じて多数の部品が提供されることはあり得る。このような状況では、部品を置き換えるという要求が自然に発生すると考えられる。例えば、ユーザが多数の部品の中から所望の部品を選べたり、機器が備えるリソース条件から動作すべき部品が決定できたりする、そのような機能を提供することは望ましいだろう。

【0014】

しかし、部品の置換を実現するのにアプリケーションの開発し直しが必要となるのでは柔軟性が高いとは言えない。多様な機器に搭載するアプリケーションの部品を機器毎に変更したい場合に、その度に開発のやり直しを行うのでは開発コストがかかり過ぎる。一方、部品の置換を行いたい別の状況が存在する。

【0015】

今日、多くのネットワークサービスが提供されている。機器がネットワークサービスと連携することで従来できなかつたサービスが実現されていることが多い。機器の付加価値向上のためにはネットワークサービスとの連携が欠かせない。

【0016】

ここで、新たなネットワークサービスが提供される状況を考えてみる。新ネットワークサービスが作られ、提供される頻度は、機器上のソフトウェアが更新される頻度の比ではない。ネットワーク上のコンピュータの性能、リソース、開発環境と、機器のそれと比べてみれば当然である。従って、機器ソフトウェアが更新されないままであるのに、ネットワークサービスが更新され、向上していく状況が常に起こり得る。

【0017】

こういった場合、機器が出荷された後、後から作られたネットワークサービスと機器を連携させることは従来は困難であった。一部の機器ではファームウェアアップデート機能が備えられる。それを使えば新ネットワークサービスと連携させることは不可能ではない。しかしながら、ファームウェアアップデートはユーザに負担をかけるし、ROM 容量等の面からソフトウェアの変更の自由度は低い。さらに、上述したのと同様に、これも開発行為が必要となる。

【0018】

本発明は、アプリケーションの開発し直しが不要で、アプリケーションの実行時に部品の置換を可能とする装置を提供する。

【課題を解決するための手段】

10

20

30

40

50

【0019】

アプリケーションを構成する部品と外部サービスとを結合する方法であって、
前記外部サービスのインターフェースの情報を取得する取得工程と、
 他の部品と結合された処理対象の部品が他の部品のサービスを利用するためのインターフェースに、前記取得工程で取得した情報が示す外部サービスのインターフェースが適合するか判断する判断工程と、
 前記判断工程によって、前記処理対象の部品が前記他の部品を利用するためのインターフェースに前記外部サービスのインターフェースが適合すると判断された場合、前記処理対象の部品が、前記取得工程で取得した情報が示す外部サービスと結合されるように、前記処理対象の部品と前記他の部品との結合を変更する変更工程と、

前記外部サービスと結合された部品を実行する実行工程と、
 を備えることを特徴とする方法。

【発明の効果】

【0020】

本発明によれば、アプリケーションの開発し直しが不要で、アプリケーションの実行時に部品の置換が可能になる。

【図面の簡単な説明】

【0021】

【図1】第1の実施形態における部品結合装置100の構成を示すブロック図。

【図2】部品と部品の結合及び部品と外部サービスとの結合を示す図。

【図3】(A)は部品結合関係をSCA仕様に則って記述した部品結合定義を示す図、(B)はalternativeタグが追加されている部品結合定義を示す図。

【図4】複数の部品及び複数の外部サービスの部品結合関係を示す図。

【図5】図4に示す部品結合関係を表現した表形式のデータを示す図。

【図6】部品置換を行う際の全体の概略処理を示すフローチャート。

【図7】部品結合処理(S602)の詳細を示すフローチャート。

【図8】部品Aと部品Bとの結合の間に代理オブジェクトが介在する状態を示す図。

【図9】部品実行処理(S604)の詳細を示すフローチャート。

【図10】部品Aが外部サービスCに結合される場合を説明するための図。

【図11】部品置換処理(S605)の詳細を示すフローチャート。

【図12】更新後の部品結合定義の一例を示す図。

【図13】第2の実施形態における部品結合処理を示すフローチャート。

【発明を実施するための形態】

【0022】

以下、図面を参照しながら発明を実施するための形態について詳細に説明する。本実施形態では、ソフトウェア部品(部品)又はコンポーネントを結合してアプリケーションを構築する部品結合装置について説明する。

【0023】

[第1の実施形態]

図1は、第1の実施形態における部品結合装置100の構成を示すブロック図である。図1に示す中央処理装置(CPU)101は、装置全体を制御すると共に、置換部110、結合部111、実行部112、出力部113、入力部114、及びインターフェース検証部115として機能する。

【0024】

ハードディスクドライブ(HDD)102は、記憶部(メモリ)であり、HDD102にはアプリケーション120が記憶されている。尚、アプリケーション120は部品結合装置100の出荷時から格納されていても良いし、出荷後にHDD102にインストールされても良い。

【0025】

アプリケーション120は、1以上の部品121と、それらの部品121の結合方法を

10

20

30

40

50

記述した部品結合定義 122 を含む。また、部品として提供されるソフトウェア以外のソフトウェア、即ち、アプリケーションロジック（不図示）もアプリケーション 120 に含まれる。尚、当然のことながら、アプリケーションロジックから部品 121 を利用することは可能である。

【0026】

アプリケーションロジックから部品 121 を利用する方法は、部品 121 から他の部品 121 を利用する方法と変わりがない。そのため、以下ではアプリケーションロジックに対して、特に言及しない。

【0027】

ネットワーク I/F 103 は、部品結合装置 100 を装置外部に存在するネットワーク 150 に接続するためのインターフェースである。ROM 104 には、各種動作プログラムが読み取り可能な形式で格納されている。RAM 105 は、CPU 101 が ROM 104 に格納されているプログラムに従って処理を実行中に使用される一時的な値等を保存するメモリである。

【0028】

Flash ROM 106 は、各種設定データファイル等を保存しておくための不揮発性メモリである。尚、不揮発性という性質が HDD 102 に近いことから Flash ROM 106 を本発明における記憶部として用いることも可能である。

【0029】

ディスプレイ 107 は、各種情報をユーザに表示する表示装置である。操作部 108 は、具体的にはキーボード、マウス、タッチディスプレイ等のユーザが入力操作を行うものである。部品結合装置 100 はネットワーク I/F 103、ネットワーク 150 を介して PC 130 及びサーバ 140 に接続されている。

【0030】

次に、上述の構成を有する部品結合装置 100 において、アプリケーション 120 の中の部品 121 間の結合及び部品 121 と外部サービスとの部品結合関係を、図 2 を用いて説明する。

【0031】

図 2 は、部品と部品の結合及び部品と外部サービスとの結合を示す図である。サービスコンポーネントアーキテクチャ（SCA）仕様によれば、部品がその機能を部品外部に公開するインターフェースはサービスと呼ばれ、部品が別の部品のサービスを利用するためのインターフェースは参照と呼ばれる。図 2 に示す例では、部品 A が 2 つの参照 210A、210B のインターフェースを有し、部品 B がサービス 220A のインターフェースを有し、外部サービス C がサービス 220B のインターフェースを有する。ここで、ある参照があるサービスを利用するときに、その参照とそのサービスとが結合される。

【0032】

図 2 では、部品 A が部品 B を利用する際の参照 210A とサービス 220A との結合を関係線 230A で示し、部品 A が外部サービス C を利用する際の参照 210B とサービス 220B との結合を関係線 230B で示している。このとき、結合された参照 210A とサービス 220A とのインターフェースを整合しなければならない。整合しなければ、参照 210A からサービス 220A の全ての機能を利用することができないからである。

【0033】

Java（登録商標）言語で言えば、参照 210A が依拠する Java（登録商標）インターフェースクラスとサービス 220A が依拠する Java（登録商標）インターフェースクラスとの互換性がとれていなければならない。

【0034】

具体的には、サービス 220A が依拠する Java（登録商標）インターフェースクラスは参照 210A が依拠する Java（登録商標）インターフェースクラスと一致しているか、もしくはスーパーセットでなければならない。非特許文献 1 には、その他の条件も記載されている。

10

20

30

40

50

【0035】

図2に示すように、結合は同一のアプリケーション120の中の部品同士だけでなく、アプリケーション120内の部品Aとアプリケーション120外の外部サービスCとの間でも行うことができる。その意味で、外部サービスCは部品A、部品Bと同様に取り扱うことができる。

【0036】

図3の(A)に、図2に示す部品結合関係をSCA仕様に則って記述した部品結合定義122の一部を示す。図3に示す(A)では、説明のために行番号が付与されているが、本来はXML言語形式で記述されるものである。

【0037】

2行目は部品Aについての記述であることを示す。3行目は部品Aの参照210Aを示し、その名前は「b」であること、それがターゲット(結合先)部品Bに結合されることを示す。4行目は参照210AのインターフェースがJava(登録商標)インターフェースクラスBBC(不図示)に従うことを示す。

【0038】

以下同様に、6~9行目は参照210Bが外部サービスCに結合されることを示す。尚8行目は結合がWebサービスを用いて行われることを示す。Webサービスでアクセスするのに必要となるURI(Uniform Resource Identifier)はuri属性で示される。

【0039】

部品Aの参照210Bと外部サービスCのサービス220Bとの間のインターフェースの整合性は次のように確認する。まず、参照210Bのインターフェースについては、7行目でそれがJava(登録商標)インターフェースクラスCCCに従うことが分かる。サービス220Bが提供するインターフェースについては、それに対応するSDLファイル(不図示)の内容を調べる。一般に、Webサービスではサービスのインターフェースの内容を示すSDLファイルが提供される。従って、SDLファイルを調べればサービス220Bのインターフェースの内容を知ることができる。それがJava(登録商標)インターフェースクラスCCCと互換性があるか否かを調べれば良い。

【0040】

次に、第1の実施形態におけるアプリケーション120内の複数の部品及び複数の外部サービスの部品結合関係を、図4を用いて説明する。ここで、アプリケーション120は複数の部品121として部品Aから部品Lを含む。

【0041】

図2では、参照とサービスとを明示していたが、図4ではそれらを省略し、任意の部品から別の部品への結合を単純な矢印で表現している。実線の矢印は実際に結合されていることを示し、破線の矢印は結合候補であることを示す。例えば、部品Aは現在、部品Bに結合されている。両者の間の結合インターフェースはインターフェースBBCである。部品Bの他にインターフェースBBCを満たす2つの部品(部品D、部品E)がアプリケーション120内に存在し、部品Aはそれらとも結合することができる。

【0042】

また、現在結合されているものが部品であれば、結合候補も部品でなければならないということはない。即ち、部品Fは部品Gに結合されているが、そのインターフェースGGGを満たす外部サービスCを結合候補とすることもできる。

【0043】

また、部品Hの場合、その結合が示すように、現在は外部サービスIに結合されているが、インターフェースIIIを満たす外部サービスJを結合候補とすることもできる。更に、部品Kは、現在部品Lに結合されているだけで、結合候補は持っていない例である。

【0044】

ここで、図4に示す部品結合関係を表現した表形式のデータを、図5を用いて説明する。この表の内容となるデータはRAM105に格納される。図5において、部品列501は参照210を持つ部品121のリストである。参照列502は、対応する部品列501

10

20

30

40

50

の部品 121 が備える参照 210 である。この例では、一つの部品につき一つの参照しか記述していないが、一つの部品につき複数の参照が存在しても良い。その場合には一つの部品につき複数の行が存在することになる。

【0045】

使用部品列 503 は、対応する参照列 502 の参照が結合している先の部品である。尚、上述したように、外部サービスはアプリケーション 120 内の部品 121 と同様に取り扱えるので、使用部品列 503 に外部サービスが入っていても良い。

【0046】

部品候補列 504 は、対応する参照列 502 の参照が結合する候補となる部品のリストである。また、部品候補列 504 の一つのセル内に複数の部品及び外部サービスが入っても良い。インターフェース列 505 は、対応する参照列 502 の参照が従うインターフェースである。

10

【0047】

次に、図 5 に示す表データを生成する基となる部品結合定義 122 を図 3 に示す (B) を用いて説明する。図 3 に示す (A) を用いて説明した、SCA 仕様で一般的な部品結合定義 122 とほぼ同じであるが、図 3 の (B) では更に、alternative タグが追加されている。このタグを用いて結合候補部品が記述される。

【0048】

例えば、5 ~ 6 行目では、参照 a に対する結合候補として部品 D と部品 E が記述されている。12 ~ 14 行目では、参照 f に対する結合候補として外部サービス C が挙げられている。これらの結合候補が部品候補列 504 に格納される。

20

【0049】

ここで、第 1 の実施形態における部品結合装置 100 が部品置換を行う際の全体の概略処理を、図 6 を用いて説明する。最初に、部品結合装置 100 が外部装置（不図示）からアプリケーション 120 をインストールする (S601)。このインストールの操作は、ユーザが操作部 108 とディスプレイ 107 を用いて行う。そして、操作部 108 からインストールが指示されると、CPU101 がネットワーク 150 に接続されている外部装置からネットワーク I/F 103 を介してアプリケーション 120 をダウンロードして HDD102 に格納する。

【0050】

30

次に、ユーザが操作部 108 からアプリケーション 120 の開始を指示すると、CPU101 がアプリケーション 120 内の部品結合定義 122 を読み出し、解析することで、アプリケーション 120 内の部品 121 の結合処理を行う (S602)。この結合処理を行った後、CPU101 はアプリケーション 120 を実際に開始する。それに伴い、部品 121 のコードを実行する (S604)。

【0051】

これ、ユーザが現在使用している部品 121 を置き換えたくなった時、部品 121 の置換処理を指示する。この指示に応じて CPU101 は部品 121 の置換処理を実行する (S605)。部品 121 の置換処理を行った後には (S606)、ループの先頭に戻り (S603)、置き換えた部品を使用する。つまり、CPU101 は置換した部品 121 のコードを実行する (S604)。

40

【0052】

尚、この例では、一旦、部品 121 を実行した後、部品 121 の置換を行っているが、アプリケーション 120 をインストールした直後、即ち、実行する前に部品 121 の置換を行っても良い。また、図 8 では省略しているが、アプリケーション 120 を停止したり、アンインストールしたりすることも当然のことながら可能である。

【0053】

次に、第 1 の実施形態における部品結合処理 (S602) の詳細を、図 7 を用いて説明する。以下のシーケンスでは結合部 111 として機能する CPU101 が部品結合処理の動作主体となる。

50

【0054】

最初に、CPU101がHDD102に格納されたアプリケーション120内の部品結合定義122を読み出す(S701)。そして、CPU101は読み出した部品結合定義122を解釈する(S702)。即ち、CPU101は図3に示す(B)のようなXML言語で記述された部品結合定義122を解釈し、図5に示すような表形式のデータを生成し、RAM105に格納する。

【0055】

部品結合定義122には、1以上の部品121についてのデータが存在する。そこで、ループ1の先頭(S703)から末尾(S709)までの処理を部品121毎に繰り返す。また、一つの部品121内には1以上の参照が存在し得る。ここで、SCAの仕様上は参照が存在しない部品121もあり得るが、本発明は部品結合を関心とするので、結合が不可能な参照がない部品121のことはここでは考えない。つまり、ループ2の先頭(S704)から末尾(S708)までの処理を参照毎に繰り返す。

10

【0056】

次に、CPU101は処理の対象である参照の結合先部品が外部サービスか否かを判定する(S705)。尚、使用部品列503について、外部サービスは部品121と同様な取り扱いが可能であると説明したが、それは両者の区別がつかないという意味ではない。S702で部品結合定義122を解釈した時に、結合先部品がアプリケーション120内の部品121(内部部品)か、外部サービスかを示す情報がRAM105に格納される。

20

【0057】

S705で、CPU101が外部サービスではなく、内部部品121と判定した場合、内部部品121用の代理オブジェクトのインスタンスを生成する(S706)。この代理オブジェクトのインスタンスとは、あるクラスのインターフェースを満たすように、動的に生成される代理クラスのインスタンスである。

【0058】

Java(登録商標)言語では、あるインターフェースクラスに対して、代理クラスをプログラム中で生成することができる。その代表的な用途は、あるインターフェースクラスを実装した実装クラスのオブジェクトのメソッドをプログラムが呼び出すときに、代理クラスがその呼び出しをキャッチし、そのタイミングで何か別の処理を行うことである。代理クラスは間接的に呼び出しを行う機能を提供する。部品結合においては、その機能が有用であるため、第1の実施形態においては代理クラスを使用している。

30

【0059】

代理クラスは、部品121の参照が従うインターフェース401に基づいて生成される。従って、S706で代理オブジェクトインスタンスを生成する直前に、代理クラスが生成されれば良い。但し、当然のことながら、図7のループ処理を実行中、同一の代理クラスから複数個の代理オブジェクトインスタンスを生成する場合は、その都度、代理クラスを生成する必要はなく、一度代理クラスを生成すれば良い。

【0060】

また、上述のS705で、結合先部品が外部サービスと判定した場合、CPU101は外部サービス用の代理オブジェクトのインスタンスを生成する(S710)。ここで代理オブジェクトのインスタンスは結合先部品に関連する情報を含むことに注意すべきである。例えば、代理オブジェクトのインスタンス生成時には、結合先の内部部品121のクラス情報或いは外部サービスのアクセス用部品(詳細は後述する)のクラス情報がコンストラクタパラメータとして渡され、インスタンス内部に保存される。これらのクラス情報は結合先内部部品121或いは外部サービスアクセス用部品のインスタンスを生成するのに用いられる。

40

【0061】

上述したS706又はS710の処理の後に、部品121の参照に対応するフィールドに代理オブジェクトインスタンスを設定する(S907)。ここでフィールドとは、部品121オブジェクトの中にある情報格納場所であり、Java(登録商標)言語で言うと

50

このフィールドに相当する。フィールドに設定されたオブジェクトを通じて、他のクラスのメソッドを呼び出すことができる。他の言語でもこのフィールドに相当するものは存在する。

【0062】

図8に、部品Aと部品Bとの結合の間に代理オブジェクトが介在する状態を示す。アプリケーション120側から見ると、アプリケーション120は代理オブジェクトBの存在を知らないので、部品Aが部品Bに直接結合されているように見える。

【0063】

しかし、CPU101は結合の間に代理オブジェクトBを介在させる。即ち、S706又はS710で代理オブジェクトBのインスタンスを生成し、S707でそれを部品Aの参照210Aに対応するフィールド1203Aに設定する。設定動作を矢印1202Aで示している。この設定により部品Aと代理オブジェクトBが結合される。

【0064】

代理オブジェクトBは、部品Aの参照210Aが従うインターフェースと整合性を持つ代理クラスから生成されたものである。従って、代理オブジェクトBのサービス220Cのインターフェースは、必然的に参照210Aのインターフェースと互換性を持ち、参照210Aとサービス220Cが結合される。つまり、部品Aは、参照210Aを通じて代理オブジェクトBのメソッドの呼び出しを確実に行うことができる。代理オブジェクトBと部品Bとの結合については後述する。

【0065】

図7の説明に戻ると、S708でループ2の末尾に至り、S704のループ先頭に戻る。S709でループ1の末尾に至り、S703のループ先頭に戻る。以上で部品結合手順を終了する。

【0066】

次に、第1の実施形態における部品実行処理(S604)の詳細を、図9を用いて説明する。以下のシーケンスでは実行部112として機能するCPU101が部品実行処理の動作主体となる。

【0067】

最初に、アプリケーション120が、部品Aが提供するサービスを呼び出す(S901)。即ち、CPU101が、部品Aが提供するサービスの実装であるメソッドを呼び出す。図8に示すように、部品Aが部品Bを使用しているものとする。部品Aは部品Bを使用するために、参照210Aが従うインターフェースのメソッドを呼び出す(S902)。

【0068】

参照210Aのインターフェースのメソッドは、フィールド1203Aを通じて代理オブジェクトBのメソッドを呼び出す(S903)。図7を用いて説明したように、部品結合処理でフィールド1203Aへの設定が済んでいるので、この処理が可能になる。

【0069】

代理オブジェクトBのメソッドは、自オブジェクトが外部サービス用か否かを判定する(S904)。S706又はS710で代理オブジェクトインスタンスを作成した時に、内部部品121用或いは外部サービス用の何れの用途で作成したかの情報を記憶しておき、その情報に基づいて判定を行う。判定の結果、内部部品121用と判定した場合、代理オブジェクトBのメソッドは、代理オブジェクトインスタンス内に格納された結合先内部部品のクラス情報を用いて結合先内部部品のインスタンスを生成する(S905)。この処理では、結合先内部部品は部品Bであるので、部品Bのインスタンスが生成される。

【0070】

また、S904で外部サービス用と判定した場合、代理オブジェクトBのメソッドは、結合先外部サービスアクセス用部品のインスタンスを生成する(S908)。このインスタンスの生成は、代理オブジェクトインスタンス内に格納された結合先外部サービスアクセス用部品のクラス情報を用いて行われる。例えば、図10に示すように、部品Aが外部サービスCに結合される場合、外部サービスCにアクセスするための外部サービスCアク

10

20

30

40

50

セス用部品のインスタンスが生成される。

【0071】

代理オブジェクトBのメソッドは、S905又はS908で生成されたインスタンスを、代理オブジェクトB内のフィールド1203Bに設定する(S906)。結合先部品が部品Bの場合の設定動作を図8の矢印1202Bに示す。結合先部品が外部サービスCの場合の設定動作を図13の矢印1202Cに示す。

【0072】

最後に、代理オブジェクトBのメソッドは、フィールド1203Bに設定されたインスタンスのメソッドを呼び出す(S1007)。インスタンスが部品Bのものである場合は、このメソッド呼び出しは部品Bのメソッド呼び出しとなる。また、インスタンスが外部サービスCアクセス用部品のものである場合は、このメソッド呼び出しは外部サービスCアクセス用部品のメソッド呼び出しとなる。外部サービスCアクセス用部品のメソッド呼び出し(不図示)は、外部サービスCにアクセスするために必要な初期化を行い、その後に実際のサービスアクセス処理、例えばWebサービス通信処理を行う。

10

【0073】

以上の処理をもって、部品Aから部品Bの呼び出し、又は部品Aから外部サービスCへのアクセスが完結する。以上が部品実行処理である。

【0074】

次に、第1の実施形態における部品置換処理(S605)の詳細を、図11を用いて説明する。以下のシーケンスでは置換部110、出力部113、入力部114、インターフェース検証部115として機能するCPU101が部品置換処理の動作主体となる。

20

【0075】

最初に、ユーザがPC130にインストールされているWebブラウザを用いて部品結合装置100が提供するWebページにアクセスする(S1101)。ここでアクセスは、HTTP(Hyper Text Transfer Protocol)プロトコルを用いて行われる。このとき、ユーザが要求したページは部品の置換機能を提供するページであり、Java(登録商標)サーブレット(出力サーブレット113)によって実現されている。ここでJava(登録商標)サーブレットは、Webページを出力する手法の一例であり、他の言語や方法を用いても差し支えない。また、Java(登録商標)サーブレットによるWebページ出力もHTTPプロトコルを用いて行われる。

30

【0076】

PC130のWebブラウザから部品置換の要求を受けて、部品結合装置100の出力サーブレット113が部品置換のためのWebページをPC130に対して送信する(S1102)。ここで部品置換のためのWebページ(不図示)のページ内容には、現在の部品結合関係と、部品結合候補が記述される。ここでは、図4及び図5において、部品Aの参照aに結合される部品を置換しようとしてWebページが要求された場合を考える。この場合、部品Aの参照aには現在部品Bが結合されており、参照aに結合され得る候補部品として部品Dと部品Eが存在することを示すWebページが送信される。典型的には、結合候補部品は選択リストとして表現されているものとする。

40

【0077】

PC130のユーザは、送信されたWebページを見て、部品Aの参照aに結合すべき部品を選択する。例えば、Webページの選択リストから部品Dを選択し、フォーム送信ボタンを押下する。これによって、選択部品を記述したデータがPC130から部品結合装置100に送られ、入力サーブレット114が部品の置換内容であるデータを受信する(S1103)。

【0078】

尚、図1に示す例では、出力サーブレット113と入力サーブレット114とを異なるものとしているが、同一のサーブレットであっても良い。その場合、部品置換内容データの存在の有無により、出力として機能すべきか、入力として機能すべきかを判断し、判断結果に応じて機能すれば良い。

50

【0079】

入力サーブレット114は、部品置換内容データを置換部110に渡す。置換部110は置換後の結合先が外部サービスか否かを判定する(S1104)。ここでは、部品置換内容データを見て、部品Aの参照aに新たに結合すべき部品が部品Dであることを知る。即ち、外部サービスではなく部品Dなので、置換部110は部品D用の代理オブジェクトインスタンスを生成する(S1112)。そして、置換部110は部品Aの参照aに対応するフィールドにその代理オブジェクトインスタンスを設定する(S1108)。

【0080】

次に、部品置換内容を反映するように、部品結合関係を示すデータ(図5の表データ)を更新する(S1109)。具体的には、部品Aの参照aに対応する使用部品列503のセルの内容を、部品Bから部品Dに置き換える。また、部品候補列504のセルの内容のうち、部品Dを部品Bに置き換える。部品Eについてはそのままとする。これによって、部品Dが使用部品となると共に、今まで使用していた部品Bは結合候補となる。

10

【0081】

続いて、部品置換内容を反映するように、部品結合定義122を更新する(S1110)。更新後の部品結合定義122の一例を図12に示す。図12と図3に示す(B)とを比較すると、3行目において、部品Aの参照aに対応する結合先部品が部品Bから部品Dに置き換えられている。また、5行目において、結合候補が部品Dから部品Bに置き換えられている。

【0082】

20

また、部品結合定義122の更新内容は、HDD102に書き込まれる。これにより、部品結合装置100が再起動された後、部品結合処理が実行されたとき、置換後の内容で部品121が結合されるという効果がある。

【0083】

最後に、置換部110が置換成功を出力サーブレット113に通知し、出力サーブレット113が置換成功通知のためのWebページをPC130に送信する(S1111)。これにより、PC130のWebブラウザ上には置換成功を示すWebページが表示され、ユーザは置換成功を知る。

【0084】

また同様に、ユーザが部品Fの参照fに結合される部品を部品Gから外部サービスCに置き換える場合を、図11を用いて説明する。S1101～S1103については上述の場合と同様である。ここで、部品Fの参照fに結合される部品として、外部サービスCをPC130のユーザは選択したとする。

30

【0085】

置換部110は、ユーザが選択した部品121を見て置換後の結合先が外部サービスか否かを判定する(S1104)。この場合、外部サービスなのでS1105へ進み、インターフェース検証部115が、指定された部品である外部サービスCのインターフェース情報を取得する。具体的には、外部サービスCに対応するWSDLファイルの内容を調べる。WSDLファイルの位置は、部品結合定義122の中に記述されている外部サービスCに対するアクセスURI情報又はそのWSDLファイル位置情報(不図示)から知ることができる。

40

【0086】

次に、インターフェース検証部115が、取得したインターフェース情報から外部サービスCのインターフェースが参照fのインターフェースに適合しているか否かを判断する(S1106)。ここで適合していると判断すれば、置換部110は外部サービスC用の代理オブジェクトインスタンスを生成する(S1107)。そして、置換部110は部品Fの参照fに対応するフィールドに、その代理オブジェクトインスタンスを設定する(S1108)。S1109以降は上述の通りである。

【0087】

これに対して、S1106で適合していないと判断すれば、置換部110は置換エラー

50

を出力サーブレット 113 に通知し、出力サーブレット 113 が置換エラー通知のための Web ページを PC130 に送信する (S1113)。これにより、PC130 の Web ブラウザ上には置換エラーを示す Web ページが表示され、ユーザは置換エラーを知ることができる。

【0088】

図 12 に示すように、10 ~ 14 行目は、部品 F の参照 f に結合される部品が部品 G から外部サービス C に置き換えられている。更に、処理は説明していないが、図 12 に示す 17 ~ 23 行目は、部品 H の参照 h に結合される部品が外部サービス I から外部サービス J に置き換えられている。

【0089】

上述した処理では、PC130 の Web ブラウザに対して結合部品候補を出力し、PC130 のユーザがその結合部品候補から結合部品を選択していたが、本発明はこれに限定されるものではない。

【0090】

S1102 で、部品置換用の Web ページを送信するとき、自由入力用のテキストフィールドを含んだ Web ページを送信する。ここで、候補にはない外部サービスの URI を PC130 のユーザがテキストフィールドに入力し、その入力内容を部品結合装置 100 に送ったとする。この場合には、S1103 でその URI が置換内容として入力される。尚、S1104 以降の処理は上述の処理と同様であり、新たに指定された外部サービスが結合される。

10

20

【0091】

このとき、S1109 と S1110 は上述の処理と少し異なる。新たに指定された外部サービスは使用部品として認識され、今まで使用していた部品 121 は結合部品の候補として認識される。従って、結合候補の部品が一つ増加する。それを示すように図 5 に示す表データと部品結合定義 122 が更新される。

【0092】

第 1 の実施形態は、結合候補部品の出力と選択した部品の入力を、HTTP プロトコルを用いて行っており、そのため、出力部 113 は出力サーブレットであり、入力部 114 は入力サーブレットであった。

【0093】

30

しかし、本発明はこれだけに限定されるものではない。結合候補部品の出力と選択した部品の入力は他の一般的な入出力デバイスを用いて行っても良い。例えば、ディスプレイ 107 と操作部 108 を用いて同様に結合候補部品の出力と選択した部品の入力が可能である。

【0094】

[第 2 の実施形態]

次に、図面を参照しながら本発明に係る第 2 の実施形態を詳細に説明する。第 1 の実施形態では、出力部 113 と入力部 114 を用いて部品 121 を置換した。第 2 の実施形態では、出力部 113 と入力部 114 を用いずに部品 121 を置換する方法を説明する。

【0095】

40

第 2 の実施形態における部品結合装置の構成は、第 1 の実施形態での図 1 に示す構成と同様であり、その説明は省略する。

【0096】

ここで、第 2 の実施形態における部品結合処理を、図 13 を用いて説明する。図 13 において、S1301 ~ S1304 は S701 ~ S704 と同様である。

【0097】

S1305 で、結合部 111 として機能する CPU101 が現在処理中の参照について使用部品と候補部品の情報を取得する。S1302 において、図 5 に示す表データを取得済みなので、そこから取得することができる。

【0098】

50

次に、結合部 111 はアプリケーション 120 のメタ情報を取得する (S1306)。ここでメタ情報とは、アプリケーション 120 に付随する情報であり、第 2 の実施形態では不図示であるが、アプリケーション 120 に内包されるものとする。例えば、Java (登録商標) アプリケーションの場合、jar ファイルの中のマニフェストファイルがメタ情報に相当する。

【0099】

メタ情報には様々な情報が格納されるが、ここでは優先使用部品情報が格納されているものとする。この優先使用部品情報とは、それが指す部品 121 が候補部品の中にあれば、使用部品として指定された部品 121 より優先して使用するものである。

【0100】

次に、結合部 111 は、取得したアプリケーション 120 のメタ情報中の優先使用部品情報を見て現在処理中の参照に対応する使用部品をそのまま使用すべきか否かを判定する (S1307)。

【0101】

現在処理中の参照に対応する使用部品が優先使用部品情報の中にあれば、そのまま使用すべきだと判断される。一方、現在処理中の参照に対応する候補部品が優先使用部品情報の中にあれば、現在の使用部品がその候補部品に置き換えられるべきだと判断される。

【0102】

ここで使用部品をそのまま使用すべきだと判断された場合は、S1309 以降の処理を実行する。S1309 ~ S1314 は S705 ~ S710 と同様である。

【0103】

また、使用部品が置き換えられるべきだと判断された場合は、現在の使用部品を優先使用部品情報の中に存在する候補部品に置換する (S1308)。具体的には、S1109 と S1110 で説明したように、表データと部品結合定義 122 が更新される。そして、S1309 以降の処理を実行する。

【0104】

尚、第 2 の実施形態では、優先使用部品情報として候補部品情報が格納されている例を示したが、代わりに新たな外部サービスに関する情報が格納されていても良い。その場合、候補にない、新たな外部サービスを使用部品として入力した上述の場合と同様の処理を行う。

【0105】

また、第 2 の実施形態では、アプリケーション 120 のメタ情報中の優先使用部品情報を用いて部品の置換が行われたが、他の方法によるものでも良い。例えば、Flash ROM 106 に優先使用部品情報が格納され、結合部 111 がそれを読み出し、使用部品の置換の可否を判断しても良い。

【0106】

優先使用部品情報そのものを使用しない実施形態も考えられる。例えば、部品結合装置 100 のスペック情報を結合部 111 が取得し、規定以上のスペックであれば、部品 D を、或いは規定以下のスペックであれば部品 B を使用するなどと判定しても良い。

【0107】

実施形態によれば、アプリケーションの開発し直しが不要で、アプリケーションの実行時に部品の置換が可能になる。また、結合候補部品を選ぶだけで部品の置換が可能であり、アプリケーションの実行時に部品の置換が簡単に行える。

【0108】

また、外部サービスを指定でき、インターフェースを検証できることでアプリケーションが意図していなかったサービスとも確実に連携できる。例えば、アプリケーションが完成した後に作成されたネットワークサービスとも連携できる。

【0109】

[他の実施形態]

また、本発明は、以下の処理を実行することによっても実現される。即ち、上述した実

10

20

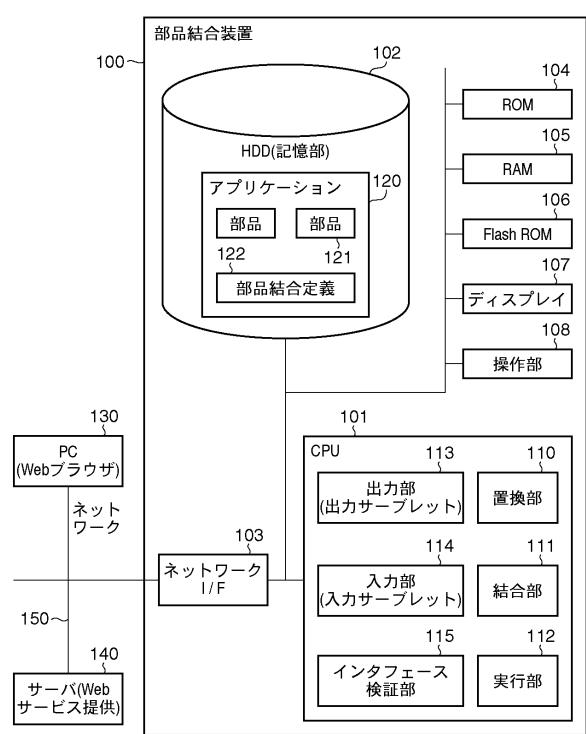
30

40

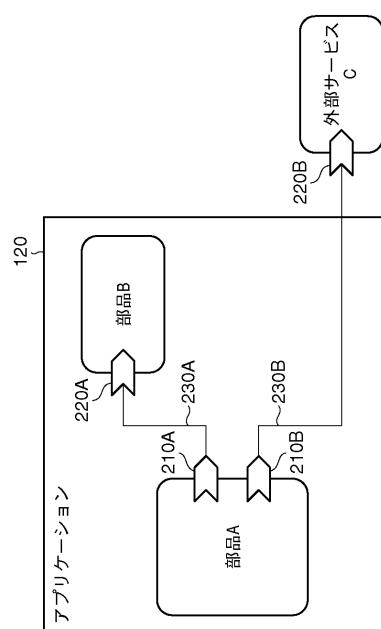
50

施形態の機能を実現するソフトウェア(プログラム)を、ネットワーク又は各種記憶媒体を介してシステム或いは装置に供給し、そのシステム或いは装置のコンピュータ(またはCPUやMPU等)がプログラムを読み出して実行する処理である。

【図1】



【図2】



【図3】

```

1: <composite>
2:   <component name="A">
3:     <reference name="b" target="B">
4:       <interface java interface="BBB">
5:     </reference>
6:     <reference name="c">
7:       <interface java interface="CCC">
8:         <binding ws url="サービスCのURI"/>
9:       </reference>
10:    </component>
11: </composite>

```

(A)

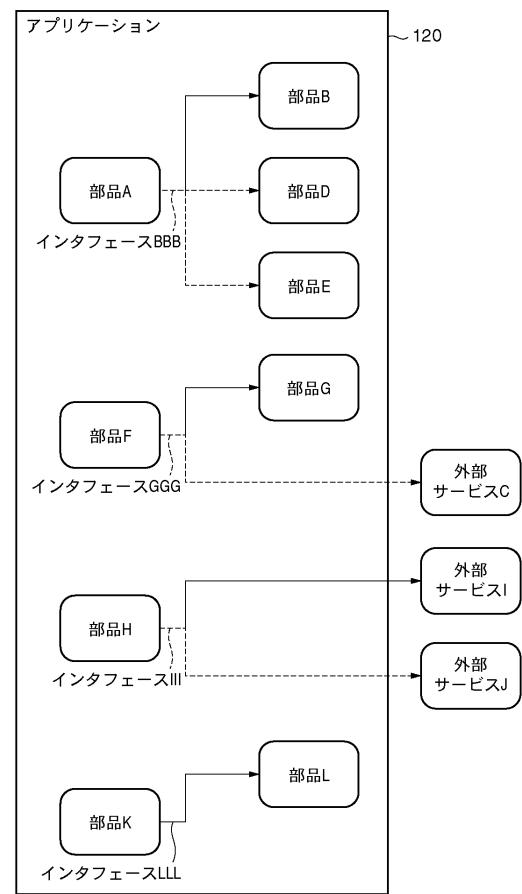
```

1: <composite>
2:   <component name="A">
3:     <reference name="a" target="B">
4:       <interface java interface="BBB">
5:     </reference>
6:     <alternative target="D"/>
7:     <alternative target="E"/>
8:   </component>
9:   <component name="F">
10:    <reference name="f" target="G">
11:      <interface java interface="GGG">
12:    </reference>
13:    <binding ws url="サービスCのURI"/>
14:  </alternative>
15:  </reference>
16: </component>
17: <component name="H">
18:   <reference name="h">
19:     <interface java interface="III">
20:     <binding ws url="サービスJのURI"/>
21:   </reference>
22:   <binding ws url="サービスJのURI"/>
23: </alternative>
24: </reference>
25: </component>
26: <component name="K">
27:   <reference name="k" target="L">
28: </component>
29: </composite>

```

(B)

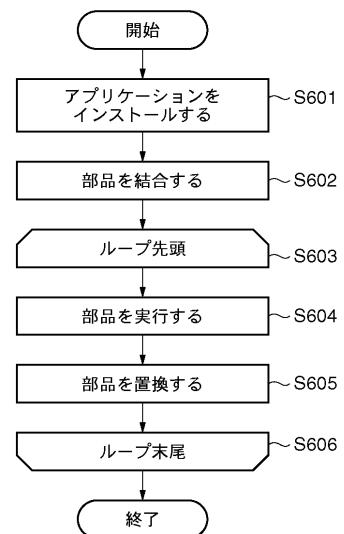
【図4】



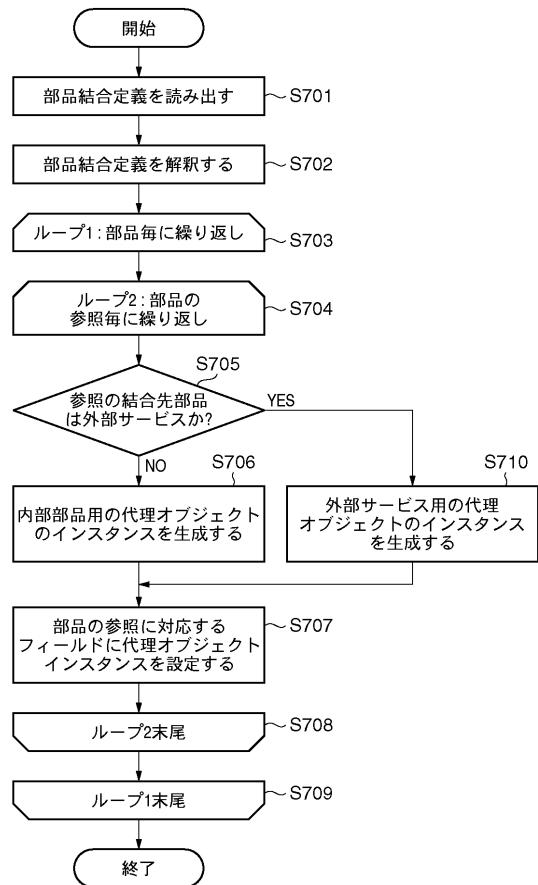
【図5】

| 部品 | 部品A | 部品B | 部品C | 部品D | 部品E | 部品F | 部品G | 部品H | 部品I | 部品J | 部品K | 部品L |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|---------|---------|---------|-----|
| 501 | 参照a | 部品B | 部品E | 部品D | 部品E | 部品F | 部品G | 部品H | 外部サービスC | 外部サービスI | 外部サービスJ | N/A |
| 502 | ～ | ～ | ～ | ～ | ～ | ～ | ～ | ～ | ～ | ～ | ～ | ～ |
| 503 | ～ | ～ | ～ | ～ | ～ | ～ | ～ | ～ | ～ | ～ | ～ | ～ |
| 504 | ～ | ～ | ～ | ～ | ～ | ～ | ～ | ～ | ～ | ～ | ～ | ～ |
| 505 | ～ | ～ | ～ | ～ | ～ | ～ | ～ | ～ | ～ | ～ | ～ | ～ |

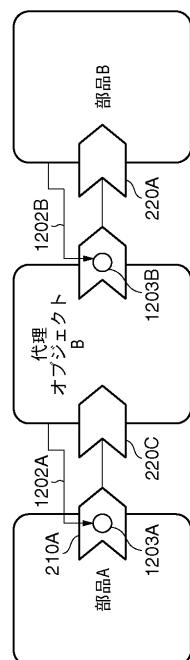
【図6】



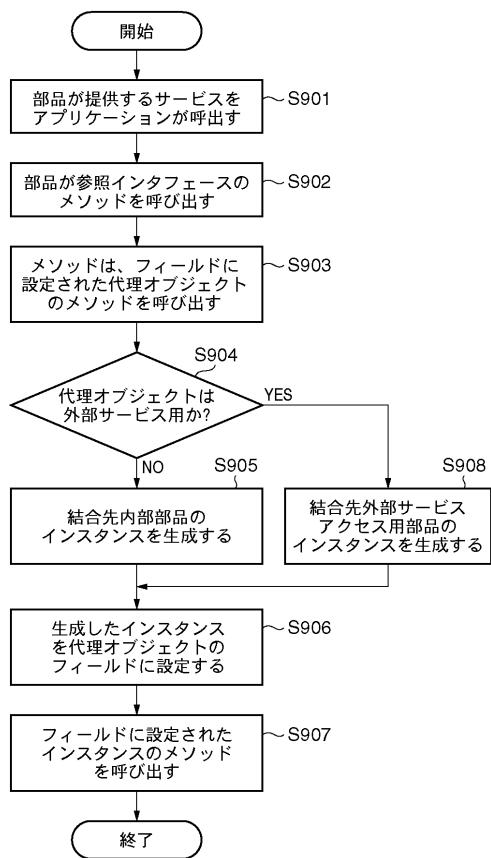
【図7】



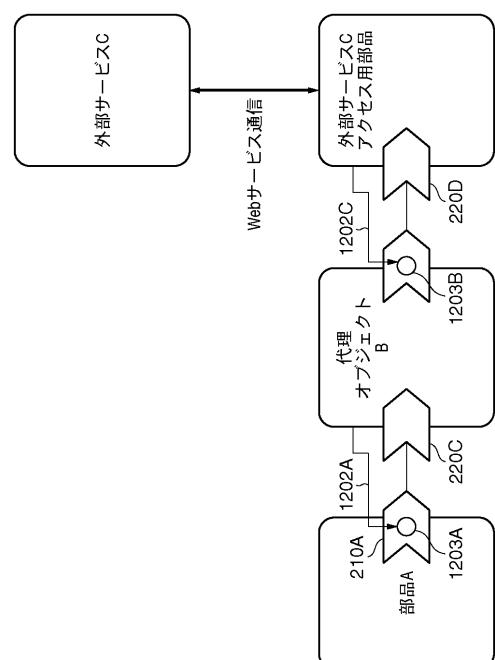
【図8】



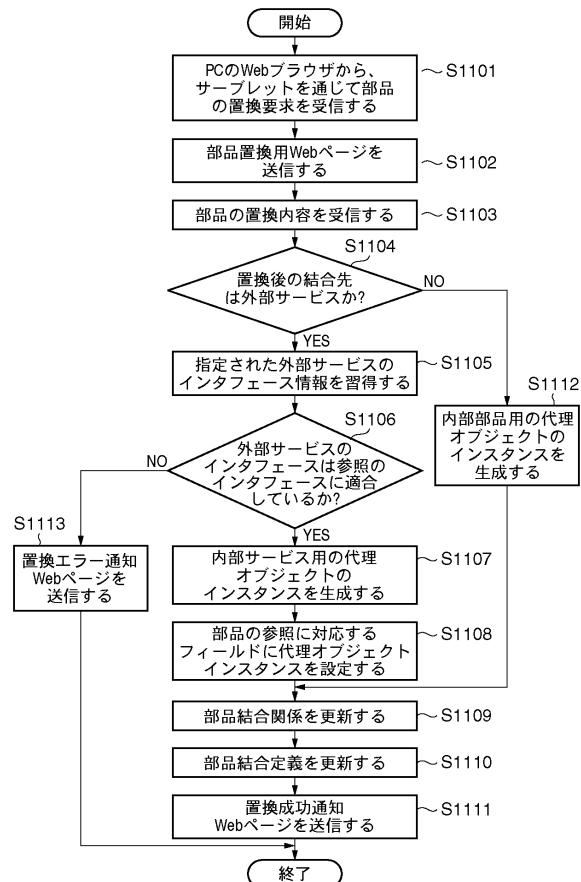
【図9】



【図10】



【図11】

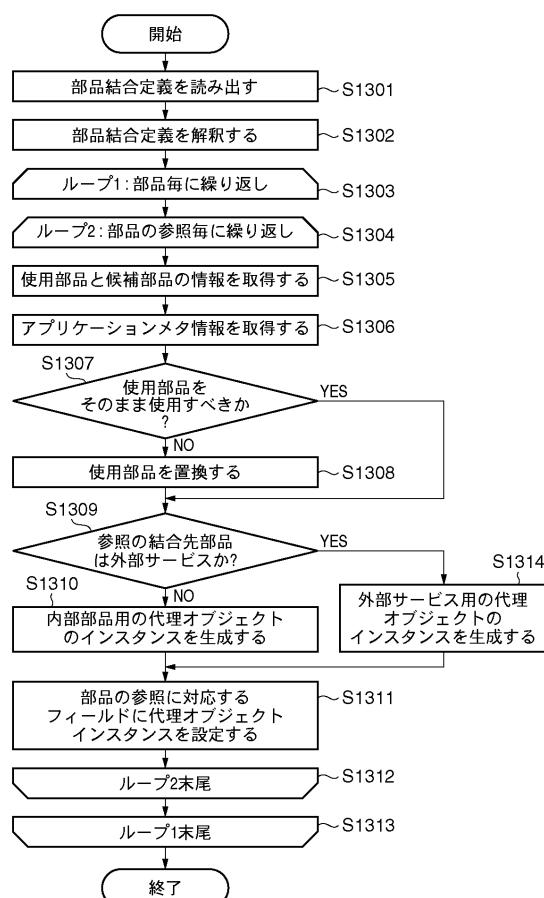


【図12】

```

1: <composite>
2:   <component name="A">
3:     <reference name="a" target="D">
4:       <interface java interface="BBB"/>
5:       <alternative target="B"/>
6:       <alternative target="E"/>
7:     </reference>
8:   </component>
9:   <component name="F">
10:    <reference name="f" target="G">
11:      <interface java interface="GGG"/>
12:      <binding ws url="サービスCのURI"/>
13:      <alternative target="H"/>
14:    </reference>
15:  </component>
16:  <component name="H">
17:    <reference name="h" target="I">
18:      <interface java interface="III"/>
19:      <binding ws url="サービスJのURI"/>
20:      <alternative target="L"/>
21:    </reference>
22:  </component>
23:  <component name="K">
24:    <reference name="k" target="L"/>
25:  </component>
26: </composite>
27: </composite>
  
```

【図13】



フロントページの続き

(72)発明者 大井 浩一

東京都大田区下丸子3丁目30番2号 キヤノン株式会社内

審査官 塚田 肇

(56)参考文献 特開2006-285313(JP, A)

特開平10-254701(JP, A)

特開2008-262351(JP, A)

Cosminexusアプリケーションサーバ V8 SOAPアプリケーション開発の手引

手引・文法書, 株式会社日立製作所, 2010年 2月, 第2版, pp. 126-129

浅井 直樹, 情報サービス産業白書2008, 2009年 5月13日, 第1版, 第95頁

(58)調査した分野(Int.Cl., DB名)

G 06 F 9 / 44