

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5613255号
(P5613255)

(45) 発行日 平成26年10月22日(2014.10.22)

(24) 登録日 平成26年9月12日(2014.9.12)

(51) Int.Cl. F I
G O 6 F 9/44 (2006.01) G O 6 F 9/06 6 2 O A

請求項の数 32 (全 18 頁)

(21) 出願番号	特願2012-540329 (P2012-540329)	(73) 特許権者	598036300
(86) (22) 出願日	平成22年7月13日 (2010.7.13)		テレフオンアクチーボラゲット エル エ
(65) 公表番号	特表2013-512488 (P2013-512488A)		ム エリクソン (パブル)
(43) 公表日	平成25年4月11日 (2013.4.11)		スウェーデン国 スtockホルム エスー
(86) 国際出願番号	PCT/EP2010/060060		1 6 4 8 3
(87) 国際公開番号	W02011/064001	(74) 代理人	100076428
(87) 国際公開日	平成23年6月3日 (2011.6.3)		弁理士 大塚 康德
審査請求日	平成25年6月13日 (2013.6.13)	(74) 代理人	100112508
(31) 優先権主張番号	61/264, 345		弁理士 高柳 司郎
(32) 優先日	平成21年11月25日 (2009.11.25)	(74) 代理人	100115071
(33) 優先権主張国	米国 (US)		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

最終頁に続く

(54) 【発明の名称】 サービスを管理するためのアプリケーションサーバ及び方法

(57) 【特許請求の範囲】

【請求項 1】

サービス（108）を管理するために少なくとも1つのリポジトリ（104、106）と相互作用するアプリケーションサーバ（102）であって、

前記サービスによってランタイム・データ（114）に実装されるアプリケーション関心事（120a、120b）のリスト（119）を識別するサービス記述（122）をロードするために前記少なくとも1つのリポジトリと相互作用するように構成された合成エンティティ（110）と、

アスペクト関心事（126a、126b）のリスト（132）を識別するアスペクト記述（128）にアクセスするために前記少なくとも1つのリポジトリと相互作用するように構成されたウィービング・エンティティ（116）であって、アスペクト関心事の前記リストは一意的なアスペクト識別情報（130）を有し、アスペクト関心事のそれぞれは一意識別子によって識別される、ウィービング・エンティティ（116）とを備え、

前記ウィービング・エンティティは、前記サービスの実行中に前記ランタイム・データ内の前記アプリケーション関心事のランタイム・サービス記述（122'）を動的に更新するように構成され、ジョインポイントにおけるアスペクト・ウィービング中に、前記ランタイム・データ内の前記アプリケーション関心事の前記ランタイム・サービス記述の更新は、前記ランタイム・データ内の前記アプリケーション関心事にウィービングされる前記アスペクト関心事のために定義された前記一意識別子を追加するか、削除するか、又は削除されたものとしてマークすることによって実行されることを特徴とするアプリケーシ

10

20

ョンサーバ。

【請求項 2】

前記ウィービング・エンティティは、前記アспект・ウィービングが前記ランタイム・データ内の前記アプリケーション関心事の 1 つに変化をもたらした場合に、当該 1 つのアプリケーション関心事を前記一意なアспект識別情報と一緒にマークするように構成されることを特徴とする請求項 1 に記載のアプリケーションサーバ。

【請求項 3】

前記ウィービング・エンティティは、2 つ以上のアспектが前記ランタイム・データ内の 1 つのアプリケーション関心事に適用可能であるならばウィービングのために前記 2 つ以上のアспектのうちの 1 つを選択し、残りのアспектをスキップするポリシーを実施するように構成されることを特徴とする請求項 1 又は 2 に記載のアプリケーションサーバ。

10

【請求項 4】

前記ウィービング・エンティティは、前記ポリシーが変わった場合に、新たなポリシーに従って前記ランタイム・データ内の前記アプリケーション関心事の前記サービス記述内の一意なアспект識別情報を更新することを特徴とする請求項 3 に記載のアプリケーションサーバ。

【請求項 5】

前記ウィービング・エンティティは、前記ランタイム・データ内のアプリケーション関心事の 1 つに対処する複数のアспектがウィービングされるならば、当該 1 つのアプリケーション関心事と一緒に複数の一意なアспект識別情報を格納するように構成されることを特徴とする請求項 3 又は 4 に記載のアプリケーションサーバ。

20

【請求項 6】

前記ウィービング・エンティティは、前記アспект関心事の 1 つが追加されるべきかを決定する場合に前記サービス記述を参照するように構成され、当該 1 つのアспект関心事の機能が前記サービスで既にカバーされているならば、当該 1 つのアспект関心事は前記ランタイム・データ内の前記アプリケーション関心事にウィービングされないことを特徴とする請求項 1 乃至 5 の何れか 1 項に記載のアプリケーションサーバ。

【請求項 7】

前記アспект関心事のリストは、アспектが前記サービスに追加するアспект関心事の個別のリストと、アспектが前記サービスから除去するアспект関心事の別の個別のリストとを含むことを特徴とする請求項 1 乃至 6 の何れか 1 項に記載のアプリケーションサーバ。

30

【請求項 8】

前記ウィービング・エンティティは、前記アспект・ウィービングが前記ランタイム・データ内の前記アプリケーション関心事の 1 つに変化をもたらす場合に前記一意な識別情報と一緒に当該 1 つのアプリケーション関心事をマークするように構成され、同じ一意なアспект識別情報を有する異なるアспект関心事が別のジョインポイントにおいてウィービングされようとしているならば、前記異なるアспект関心事が前記同じ一意なアспект情報と一緒に記述されているのでこれを許可することを特徴とする請求項 1 乃至 7 の何れか 1 項に記載のアプリケーションサーバ。

40

【請求項 9】

前記一意識別子はタグ又はキーワードを含むことを特徴とする請求項 1 乃至 8 の何れか 1 項に記載のアプリケーションサーバ。

【請求項 10】

サービス (108) を管理するために少なくとも 1 つのリポジトリ (104、106) と相互作用するアプリケーションサーバ (102) によって実施される方法であって、

前記サービスによってランタイム・データ (114) に実装されるアプリケーション関心事 (120a、120b) のリスト (119) を識別するサービス記述 (122) をロードするために前記少なくとも 1 つのリポジトリと相互作用する工程と、

50

アスペクト関心事（126a、126b）のリスト（132）を識別するアスペクト記述（128）にアクセスするために前記少なくとも1つのリポジトリと相互作用する工程であって、アスペクト関心事の前記リストは一意なアスペクト識別情報（130）を有し、アスペクト関心事のそれぞれは一意識別子によって識別される、工程と、

前記サービスの実行中に前記ランタイム・データ内の前記アプリケーション関心事のランタイム・サービス記述（122'）を動的に更新する工程とを有し、

ジョインポイントにおけるアスペクト・ウィービング中に、前記ランタイム・データ内の前記アプリケーション関心事の前記ランタイム・サービス記述の更新は、前記ランタイム・データ内の前記アプリケーション関心事にウィービングされる前記アスペクト関心事のために定義された前記一意識別子を追加するか、削除するか、又は削除されたものとしてマークすることによって実行されることを特徴とする方法。

10

【請求項11】

前記アスペクト関心事のリストは、アスペクトが前記サービスに追加するアスペクト関心事の個別のリストと、アスペクトが前記サービスから除去するアスペクト関心事の別の個別のリストとを含むことを特徴とする請求項10に記載の方法。

【請求項12】

前記動的に更新する工程は、前記アスペクト・ウィービングが前記ランタイム・データ内の前記アプリケーション関心事の1つに変化をもたらした場合に、当該1つのアプリケーション関心事を前記一意なアスペクト識別情報と一緒にマークする工程を含むことを特徴とする請求項10又は11に記載の方法。

20

【請求項13】

前記動的に更新する工程は、2つ以上のアスペクトが前記ランタイム・データ内の1つのアプリケーション関心事に適用可能であるならばウィービングのために前記2つ以上のアスペクトのうちの1つを選択し、残りのアスペクトをスキップするポリシーを実施する工程を含むことを特徴とする請求項10乃至12の何れか1項に記載の方法。

【請求項14】

前記ポリシーが変わった場合に、新たなポリシーに従って前記ランタイム・データ内の前記アプリケーション関心事の前記サービス記述内の一意なアスペクト識別情報を更新することを特徴とする請求項13に記載の方法。

【請求項15】

前記動的に更新する工程は、前記ランタイム・データ内のアプリケーション関心事の1つに対処する複数のアスペクトがウィービングされるならば、当該1つのアプリケーション関心事と一緒に複数の一意なアスペクト識別情報を格納する工程を含むことを特徴とする請求項13又は14に記載の方法。

30

【請求項16】

前記動的に更新する工程は、前記アスペクト・ウィービングが前記ランタイム・データ内の前記アプリケーション関心事の1つに変化をもたらす場合に前記一意な識別情報と一緒に当該1つのアプリケーション関心事をマークする工程を含み、同じ一意なアスペクト識別情報を有する異なるアスペクト関心事が別のジョインポイントにおいてウィービングされようとしているならば、前記異なるアスペクト関心事が前記同じ一意なアスペクト情報と一緒に記述されているのでこれを許可することを特徴とする請求項10乃至15の何れか1項に記載の方法。

40

【請求項17】

前記動的に更新する工程は、前記アスペクト関心事の1つが追加されるべきかを決定する場合に前記サービス記述を参照する工程を含み、当該1つのアスペクト関心事の機能が前記サービスで既にカバーされているならば、当該1つのアスペクト関心事は前記ランタイム・データ内の前記アプリケーション関心事にウィービングされないことを特徴とする請求項10乃至16の何れか1項に記載の方法。

【請求項18】

前記一意識別子はタグ又はキーワードを含むことを特徴とする請求項10乃至17の何

50

れか1項に記載の方法。

【請求項19】

サービス(108)を管理するためのシステム(100)であって、
少なくとも1つのリポジトリ(104、106)と、アプリケーションサーバ(102)とを備え、

前記少なくとも1つのリポジトリ(104、106)は、前記サービスによって実装されるアプリケーション関心事(120a、120b)のリスト(119)を識別するサービス記述(122)を格納し、

前記少なくとも1つのリポジトリは、アスペクトによって実装されるアスペクト関心事(126a、126b)のリスト(132)を識別するアスペクト記述(128)も格納し、アスペクト関心事の前記リストは一意なアスペクト識別情報を有し、アスペクト関心事のそれぞれは一意識別子によって識別され、

前記アプリケーションサーバ(102)は、

前記サービスによってランタイム・データ(114)に実装されるアプリケーション関心事の前記リストを識別する前記サービス記述(122)をロードするように構成された合成エンティティ(110)と、

アスペクト関心事のリストを識別するアスペクト記述にアクセスするために前記少なくとも1つのリポジトリと相互作用するように構成されたウィービング・エンティティ(116)であって、アスペクト関心事の前記リストは一意なアスペクト識別情報を有し、アスペクト関心事のそれぞれは一意識別子によって識別される、ウィービング・エンティティ(116)とを備え、

前記ウィービング・エンティティは、前記サービスの実行中に前記ランタイム・データ内の前記アプリケーション関心事のランタイム・サービス記述(122')を動的に更新するように構成され、ジョインポイントにおけるアスペクト・ウィービング中に、前記ランタイム・データ内の前記アプリケーション関心事の前記ランタイム・サービス記述の更新は、前記ランタイム・データ内の前記アプリケーション関心事にウィービングされる前記アスペクト関心事のために定義された前記一意識別子を追加するか、削除するか、又は削除されたものとしてマークすることによって実行されることを特徴とするシステム(100)。

【請求項20】

前記ウィービング・エンティティは、前記アスペクト・ウィービングが前記ランタイム・データ内の前記アプリケーション関心事の1つに変化をもたらした場合に、当該1つのアプリケーション関心事を前記一意なアスペクト識別情報と一緒にマークするように構成されることを特徴とする請求項19に記載のシステム。

【請求項21】

前記ウィービング・エンティティは、2つ以上のアスペクトが前記ランタイム・データ内の1つのアプリケーション関心事に適用可能であるならばウィービングのために前記2つ以上のアスペクトのうちの1つを選択し、残りのアスペクトをスキップするポリシーを実施するように構成されることを特徴とする請求項19又は20に記載のシステム。

【請求項22】

前記ウィービング・エンティティは、前記ポリシーが変わった場合に、新たなポリシーに従って前記ランタイム・データ内の前記アプリケーション関心事の前記サービス記述内の一意なアスペクト識別情報を更新することを特徴とする請求項21に記載のシステム。

【請求項23】

前記ウィービング・エンティティは、前記ランタイム・データ内のアプリケーション関心事の1つに対処する複数のアスペクトがウィービングされるならば、当該1つのアプリケーション関心事と一緒に複数の一意なアスペクト識別情報を格納するように構成されることを特徴とする請求項21又は22に記載のシステム。

【請求項24】

前記ウィービング・エンティティは、前記アスペクト関心事の1つが追加されるべきか

10

20

30

40

50

【請求項 25】

【請求項 26】

【請求項 27】

【請求項 28】

前記アプリケーションサーバが前記サービスの実行中にランタイム・データ内のアプリケーション関心事のランタイム・サービス記述（１２２'）を動的に更新できるように、前記アспект記述へのアクセスを提供するためのインタフェースとを備え、

【請求項 29】

【請求項 30】

前記アプリケーションサーバが前記サービスの実行中にランタイム・データ内のアプリケーション関心事のランタイム・サービス記述（１２２'）を動的に更新できるように、前記アспект記述へのアクセスを提供する工程とを有し、

ジョインポイントにおけるアспект・ウィービング中に、前記ランタイム・データ内の前記アプリケーション関心事の前記ランタイム・サービス記述の更新は、前記ランタイム

ム・データ内の前記アプリケーション関心事にウィービングされる前記アспект関心事のために定義された前記一意識別子を追加するか、削除するか、又は削除されたものとしてマークすることによって実行されることを特徴とする方法。

【請求項 3 1】

前記一意識別子はタグ又はキーワードを含むことを特徴とする請求項 3 0 に記載の方法。

【請求項 3 2】

コンピュータで実行された場合に前記コンピュータに請求項 1 0 乃至 1 8、3 0 及び 3 1 の何れか 1 項に記載の方法の各工程を実行させるコンピュータ読み取り可能コードを有するコンピュータプログラム。

【発明の詳細な説明】

【技術分野】

【0 0 0 1】

本発明はユーザへ提供されるサービス（例えば、通信サービス）を管理するためのアプリケーションサーバ、リポジトリ装置、コンピュータプログラム及び方法に関する。

【背景技術】

【0 0 0 2】

通信ネットワークでは、ユーザへ提供されるサービスをアプリケーションサーバが管理することが一般的である。サービス自体はしばしば、ユーザに所定の機能を提供するソフトウェアの部品の組み合わせである。機能は通常、サービスの開発者（例えば、ネットワークオペレータ）によって事前に規定されており、サービスの実装が変更されない限り、静的なままである。しかしながら、開発者がサービスを事前に規定した後にサービスの実装を変更したいことも一般的である。サービスを変更する 1 つの方法は、元のサービスの対象アプリケーションコードを直接に修正することなく、元のサービスに機能を追加することによってサービスの実装を変更することを可能にする技術であるアспект指向プログラミング（AOP）を用いることである。AOP は当業者に周知な技術であるが、サービスに機能を追加することをより容易にするだろう特徴の改良方法を説明する前に、AOP の基礎について簡単な説明が以下に提供される。

【0 0 0 3】

AOP は複数の機能をアプリケーションに実装することを支援するソフトウェア設計技術である。しばしば、これらの機能のコードは、1 つの機能に特化したコード部品が他の機能のコード内の複数の場所に配置される必要があるという意味で、非常に入り組んでいる。これらの特性は関心事の横断と呼ばれる。しかしながら、これらの複数の機能を専用のカプセル化された単位に分割することによってこれらのクリーン且つモジュール化した実装はしばしば不可能であるか、極端に困難である。

【0 0 0 4】

AOP はアプリケーションのコードを横断的機能のコードから分離したままにすることを可能にする。この分離した実装はアспектと呼ばれる。アспектによって実装される機能を追加するためにアプリケーションにアспектを適用することはウィービングと呼ばれる。アспектのウィービングはオフラインで行うことが可能であり、よってアプリケーションの実行前に行うことが可能である。これは通常、ポイントカットによって参照されるコード位置にアспектコードを差し込むことによって、ウィービングが対象アプリケーションから新たなアプリケーションを生成することを意味する。さらに、アспектのウィービングはオンラインで行うことも可能であり、これはアプリケーションの実行が監視され、ポイントカットによって特定される状況に達すると、アプリケーションの実行を継続する前にアспектが実行されることを意味する。

【0 0 0 5】

ウィービングはポイントカットによって誘導される。ポイントカットは、アспектコードが追加されるべき場所であるアプリケーション内のジョインポイントとして知られる位置の包括的定義である。例えば、ポイントカットは所定のメソッドのすべてのコールに

10

20

30

40

50

においてアスペクトを実行することを特定してもよい。ポイントカットは対象アプリケーションのソースコード内のポイントをキャプチャする状況に基づくか、アプリケーション実行を伴うイベントを参照する。原則として、この技術はアプリケーションを分析しなかったとしても所定の機能の自動適用を自動的に可能に示う。例えば、新たなアプリケーションがサービス実行環境内に展開されるならば、この環境内に存在するアスペクトはこの新たな対象アプリケーションに自動的に適用されうるだろう。これは特に、サードパーティのサービスプロバイダからのサービスを使用する際に、通信分野で関心がある。

【発明の概要】

【発明が解決しようとする課題】

【0006】

これらのサードパーティのサービスはネットワークオペレータによって要求される機能のすべてをサポートしないかもしれない。例えば、アクティビティ・ロギングや課金がサードパーティのサービスで実装されないかもしれない。よって、AOPを利用して、これらの特徴が対象アプリケーションに自動的に追加されうる。これを達成しうる装置及び方法が本発明の目的である。

【課題を解決するための手段】

【0007】

本出願の独立請求項にアプリケーションサーバ、リポジトリ装置、システム、コンピュータプログラム及び方法が記載される。従属請求項にアプリケーションサーバ、リポジトリ装置、システム、コンピュータプログラム及び方法の有利な実施形態が記載される。

【0008】

1つの側面では、本発明はサービスを管理するために少なくとも1つのリポジトリと相互作用するアプリケーションサーバを提供する。前記アプリケーションサーバは、(a) 前記サービスによってランタイム・データに実装されるアプリケーション関心事のリストを識別するサービス記述をロードするために前記少なくとも1つのリポジトリと相互作用するように構成された合成エンティティと、(b) アスペクト関心事のリストを識別するアスペクト記述にアクセスするために前記少なくとも1つのリポジトリと相互作用するように構成されたウィービング・エンティティであって、アスペクト関心事の前記リストは一意的なアスペクト識別情報を有し、アスペクト関心事のそれぞれは一意識別子（例えば、タグ又はキーワード）によって識別される、ウィービング・エンティティとを備え、(c) 前記ウィービング・エンティティは、前記サービスの実行中に前記ランタイム・データ内の前記アプリケーション関心事の前記サービス記述を動的に更新するように構成され、ジョインポイントにおけるアスペクト・ウィービング中に、前記ランタイム・データ内の前記アプリケーション関心事の前記ランタイム・サービス記述の更新は、前記ランタイム・データ内の前記アプリケーション関心事にウィービングされる前記アスペクト関心事のために定義された前記一意識別子を追加するか、削除するか、又は削除されたものとしてマークすることによって実行される。この利点は、アスペクトが必要とされるサービスにのみアスペクトがウィービングされることをアプリケーションサーバが確かめることである。

【0009】

さらに別の側面では、本発明はサービスを管理するために少なくとも1つのリポジトリと相互作用するアプリケーションサーバによって実施される方法を提供する。前記方法は、(a) 前記サービスによってランタイム・データに実装されるアプリケーション関心事のリストを識別するサービス記述をロードするために前記少なくとも1つのリポジトリと相互作用する工程と、(b) アスペクト関心事のリストを識別するアスペクト記述にアクセスするために前記少なくとも1つのリポジトリと相互作用する工程であって、アスペクト関心事の前記リストは一意的なアスペクト識別情報を有し、アスペクト関心事のそれぞれは一意識別子（例えば、タグ又はキーワード）によって識別される、工程と、(c) 前記サービスの実行中に前記ランタイム・データ内の前記アプリケーション関心事の前記サービス記述を動的に更新する工程とを有し、ジョインポイントにおけるアスペクト・ウィー

10

20

30

40

50

ビング中に、前記ランタイム・データ内の前記アプリケーション関心事の前記ランタイム・サービス記述の更新は、前記ランタイム・データ内の前記アプリケーション関心事にウィーピングされる前記アспект関心事のために定義された前記一意識別子を追加するか、削除するか、又は削除されたものとしてマークすることによって実行される。この利点は、アспектが必要とされるサービスにのみアспектがウィーピングされることをアプリケーションサーバが確かめることである。

【0010】

なおも別の側面では、本発明はサービスを管理するためのシステムを提供する。前記システムは、少なくとも1つのリポジトリと、アプリケーションサーバとを備え、(a)前記少なくとも1つのリポジトリは、前記サービスによって実装されるアプリケーション関心事のリストを識別するサービス記述を格納し、(b)前記少なくとも1つのリポジトリは、アспектによって実装されるアспект関心事のリストを識別するアспект記述も格納し、アспект関心事の前記リストは一意なアспект識別情報を有し、アспект関心事のそれぞれは一意識別子(例えば、タグ又はキーワード)によって識別され、(c)前記アプリケーションサーバは、(i)前記サービスによってランタイム・データに実装されるアプリケーション関心事の前記リストを識別する前記サービス記述をロードように構成された合成エンティティと、(ii)アспект関心事のリストを識別するアспект記述にアクセスするために前記少なくとも1つのリポジトリと相互作用するように構成されたウィーピング・エンティティであって、アспект関心事の前記リストは一意なアспект識別情報を有し、アспект関心事のそれぞれは一意識別子によって識別される、ウィーピング・エンティティとを備え、(iii)前記ウィーピング・エンティティは、前記サービスの実行中に前記ランタイム・データ内の前記アプリケーション関心事の前記サービス記述を動的に更新するように構成され、ジョインポイントにおけるアспект・ウィーピング中に、前記ランタイム・データ内の前記アプリケーション関心事の前記ランタイム・サービス記述の更新は、前記ランタイム・データ内の前記アプリケーション関心事にウィーピングされる前記アспект関心事のために定義された前記一意識別子を追加するか、削除するか、又は削除されたものとしてマークすることによって実行される。この利点は、アспектが必要とされるサービスにのみアспектがウィーピングされることをアプリケーションサーバが確かめることである。

【0011】

さらに別の側面では、本発明はサービスを提供するアプリケーションサーバと相互作用するリポジトリを提供する。前記リポジトリは、前記サービスによって実装されるアプリケーション関心事のリストを識別するサービス記述を格納するための記憶装置を備え、アプリケーション関心事のそれぞれは一意識別子(例えば、タグ又はキーワード)によって識別される。

【0012】

さらに別の側面では、本発明はサービスを提供するアプリケーションサーバと相互作用するリポジトリによって実施される方法を提供する。前記方法は、前記サービスによって実装されるアプリケーション関心事のリストを識別するサービス記述を格納する工程を有し、アプリケーション関心事のそれぞれは一意識別子によって識別される。

【0013】

さらに別の側面では、本発明は、コンピュータで実行された場合に前記コンピュータに、サービスによって実装されるアプリケーション関心事のリストを識別するサービス記述を格納する工程を実行させるコンピュータ読み取り可能コードを有するコンピュータプログラムを提供し、アプリケーション関心事のそれぞれは一意識別子(例えば、タグ又はキーワード)によって識別される。

【0014】

なおも別の側面では、本発明は、コンピュータで実行された場合に前記コンピュータに、(a)前記サービスによってランタイム・データに実装されるアプリケーション関心事のリストを識別するサービス記述をロードするために前記少なくとも1つのリポジトリと

相互作用する工程と、(b) アスペクト関心事のリストを識別するアスペクト記述にアクセスするために前記少なくとも1つのリポジトリと相互作用する工程であって、アスペクト関心事の前記リストは一意的なアスペクト識別情報を有し、アスペクト関心事のそれぞれは一意識別子(例えば、タグ又はキーワード)によって識別される、工程と、(c) 前記サービスの実行中に前記ランタイム・データ内の前記アプリケーション関心事の前記サービス記述を動的に更新する工程とを実行させるコンピュータ読み取り可能コードを有するコンピュータプログラムを提供し、ジョインポイントにおけるアスペクト・ウィービング中に、前記ランタイム・データ内の前記アプリケーション関心事の前記ランタイム・サービス記述の更新は、前記ランタイム・データ内の前記アプリケーション関心事にウィービングされる前記アスペクト関心事のために定義された前記一意識別子を追加するか、削除するか、又は削除されたものとしてマークすることによって実行される。

10

【0015】

本発明の更なる側面が以下の詳細な説明、図面及び任意の請求項において部分的に説明され、詳細な説明から部分的に導出され、又は本発明の実施によって習得されうる。前述の総括的な説明及び以下の詳細な説明は例示のため、および説明のためであり、開示される本発明の限定するものではないことが理解されるだろう。

【0016】

本発明のより完全な理解は添付の図面と合わせて以下の詳細な説明を参照することによって得られうる。

【図面の簡単な説明】

20

【0017】

【図1】本発明の実施形態に従って一緒にサービスを管理するアプリケーションサーバ、サービス・リポジトリ及びアスペクト・リポジトリを含む例示のシステムを説明するブロック図である。

【図2】、

【図3】本発明の実施形態に従って実行される例示のウィービング・シナリオを説明するのに役立つ2つの図である。

【発明を実施するための形態】**【0018】**

AOPを利用して元のサービスの対象アプリケーションコードを直接に修正せずに元のサービスに機能を自動的に追加できるアプリケーションサーバが以下に記載される。アプリケーションサーバは特に、所定のアプリケーションに機能を自動的に追加する場合に、所定のアプリケーションコード内にどのフィーチャがまだ欠けているかと、どれが既にサポートされているかとを自動的に検出する。例えば、対象アプリケーションが既に課金をサポートしているならば、課金サポートを追加するアスペクトは重複又は矛盾する課金を避けるために適用されないだろう。これを実現するために、アプリケーションサーバは、アプリケーションコード及び現在適用されているアスペクトによってどの機能が既にカバーされているかを制御するためにアプリケーションのフィーチャ及び能力を動的に記述するように構成される。アプリケーションは、アプリケーションが既に対処しているフィーチャ及び関心事に対応する一意識別子、例えばタグやキーワードの集合で記述される。また、アスペクトが対処するだろう関心事も記述される。アスペクトが対象アプリケーションコードにウィービングされるならば、アスペクトによって適用される変更を反映するように、現在対処されている関心事のリストが更新される。アスペクトが自動的に追加されるべきかを決定する場合に、アスペクトが提供する機能がアプリケーションによって既にカバーされているかどうか、そしてそれ故、アスペクトが不要であるかどうかを確認するためにこのリストが参照されうる。

30

40

【0019】

上述のようなサービスを管理するように構成されたアプリケーションサーバは典型的に、1つ以上のリポジトリ装置と相互作用し、アプリケーションサーバとリポジトリ装置(群)との両方が(詳細に後述される)以下の特徴のうちの1つ以上を有するだろう。

50

【0020】

1. サービスが実装する関心事の記述は、特定のドメインからの使用のためにサービスを公開する（例えば）サービス・リポジトリ内に格納されたサービス記述に規定される（図1参照）。例えば、サービス記述はサービスのバインド情報を含んでもよく、通常はサービスの機能及び能力を記述する。ウェブサービス記述言語（WSDL）はこのようなサービス記述に用いられうる1つの方式である。

【0021】

1. 1 関心事のサービス記述は、対処される関心事ごとに1つの一意識別子（例えば、キーワード／タグ）を有する。

【0022】

2. アスペクトが実装する関心事はアスペクトの記述内に規定される。この記述はアスペクト・リポジトリ内に格納される（図1参照）。アスペクト記述は、適用されるならばどの機能をアスペクトがアプリケーションへ追加する（又はアプリケーションから削除する）かを示す。

【0023】

2. 1 関心事のアスペクト記述は、対処される関心事ごとに1つの一意識別子（例えば、キーワード／タグ）を有する。

【0024】

2. 2 関心事のアスペクト記述は、アスペクトがこの関心事の処理を追加又は削除するかを付加的に表現できる。

【0025】

2. 3 一緒になって関心事を実装する複数のアスペクトは共通ID（アスペクト・グループID）と一緒にアスペクト・リポジトリ内に登録される。

【0026】

3. アプリケーションのインスタンスが対処する関心事のリストはアプリケーションサーバ内でランタイムに動的に維持され更新される（図1参照）。例えば、アスペクトの使用はアプリケーション全体によって対処される関心事における変更を生じさせうる。

【0027】

3. 1 デフォルトのサービス関心事記述はサービス・リポジトリからアプリケーションサーバにコピーされ、サービス開始時のサービス・インスタンスでランタイム・データに格納される。

【0028】

3. 2 ランタイム・データ内のサービス関心事記述はサービスの実行を通じて更新され、例えばアスペクト・ウィービングによってこの実行中のインスタンスへ機能が動的に追加又は削除される。

【0029】

3. 3 ランタイム関心事記述の更新は、アプリケーションコードにウィービングされるアスペクトについて定義される関心事タグを追加又は削除することによって実行される。

【0030】

3. 4 アスペクトが機能を削除するならば、それぞれのキーワード／タグの削除によって、ランタイム・サービス関心事記述においてこれが反映される。

【0031】

3. 5 アスペクトが機能を削除するならば、削除されたものとして関心事をマークする情報を付加することによってランタイム・サービス関心事記述においてこれが反映される。

【0032】

3. 6 同じ関心事についてどのアスペクトが適用されるかを判定するポリシーが変わるならば、この関心事と一緒に登録されているアスペクトIDは変更／更新されてもよい。

10

20

30

40

50

【0033】

3. 7 複数のアスペクトが単一の関心事に対処するものにウィービングされるならば、ランタイム関心事記述において複数のアスペクトIDがこの関心事と一緒に格納される。

【0034】

3. 8 関心事が複数のアスペクト（アスペクトのグループ）によって実装されるならば、これらの関心事は共通グループIDと一緒に登録される。このアスペクト・グループIDは、グループ内の任意のアスペクトをウィービングした後にランタイム・サービス関心事記述内の関心事で格納される。

【0035】

4 ランタイム・サービス関心事記述は、アスペクトがアプリケーションにウィービングされるべきかどうかの決定が必要な場合に評価される。アスペクト・ウィービングは、アプリケーションがアスペクトと同じ関心事に既に対処しているならば、拒絶されうる。

【0036】

4. 1 アスペクトの関心事記述が、ランタイム・サービス関心事記述にも含まれる関心事を含むならば、アスペクト・ウィービングは拒絶されうる。

【0037】

4. 2 ウィービングにおいて、ランタイム・サービス関心事記述が、アスペクト関心事記述にも含まれる関心事を既に含むならば、対立する関心事がアスペクトの以前のウィービングによって追加された場合に、ウィービングは許可されうる。

【0038】

4. 3 ジョインポイントにおいて同じ関心事について複数のアスペクトが適用可能であるならば、どのアスペクトを適用するかに関する決定をサポートするためにポリシーが用いられうる。この観点で、複数のアスペクトは同じ関心事を実装するものの、ポリシーは複数のアスペクトがウィービングされることを可能にする。

【0039】

4. 4 結合ポイントにおいて同じ関心事について複数のアスペクトが適用可能であるならば、アスペクトの適用の順序に関して決定するためにポリシーが用いられうる。

【0040】

5 合成サービスの関心事のデフォルトのリストは合成に用いられている構成サービスのリストから導出されてもよい。このようにアスペクト・ウィービングと合成との両方が、対処される関心事のリストを動的に変更してもよい。

【0041】

図1を参照して、本発明の実施形態に従うサービス108と一緒に管理するアプリケーションサーバ102、サービス・リポジトリ104及びアスペクト・リポジトリ106を含む例示のシステム100を説明するブロック図が示される。アプリケーションサーバ102は当該分野において周知の多くのコンポーネントを含むが、これらは明確化のために図1には示されず、本発明に関連する合成エンティティ110、分析機能112、ランタイム・データ114及びウィービング・エンティティ116のようなコンポーネントが示される。これらのコンポーネント102、104、106、108、110、112、114、116のそれぞれの機能に関する詳細な説明は以下のトピックに関して以下に記載される。(1) アプリケーション及びアスペクトの関心事(特徴1. x、特徴2. x)。(2) ランタイムでの関心事処理(特徴3. x)。(3) ウェービング決定に影響する関心事(特徴4. x)。(4) その他の実施形態(特徴5)。(5) 例示のシナリオ(図2、図3)。

【0042】

1. アプリケーション及びアスペクトの関心事(特徴1. x、特徴2. x)

サービス108の機能全体は、アプリケーション全体の相異なる部分を提供する複数のサブ機能で構成される。これは特に、各構成サービス121a、121b(servicel, servicel)がアプリケーションの一部を提供する合成サービス118である場合である。合

10

20

30

40

50

成サービス118によって実装される機能は関心事120a、120bと呼ばれる。この観点でAOPは合成サービス118によってどの関心事120a、120bが対処されるかを修正する方法である。

【0043】

AOP技術の使用前の合成サービス118は複数の関心事120a、120bに対処する。例えば、合成サービス118はサービスの課金を扱うかもしれないが、合成サービス118はロギングのサポートを含まないかもしれない。本発明の特徴は、アプリケーションの実装によって既にカバーされている関心事120a、120bのリスト119を、サービス・リポジトリ104内のサービス記述112の一部として追加することである（特徴1）。1つの実施形態では、合成サービス118が対処する各関心事120a、120bはキーワード／タグ（例えば、concern1、concern2）に対応しうる。その後、関心事120a、120bの集合全体はこれらのキーワード／タグのリスト119によって表される（特徴1.1）。 10

【0044】

また、アスペクト124は関心事126a、126bの実装を提供する。これは、アスペクト・リポジトリ106内のアスペクト記述128によって反映される。アスペクト記述128は一意ID130と、アスペクト124に関連する関心事126a、126bのリスト132とを含む（特徴2）。さらに、アスペクト記述128は、これが適用されるならば、関心事126a、126bがアプリケーションへどの機能を追加するか、又はアプリケーションからどの機能を削除するかを示す。1つの実施形態では、アスペクト124が対処する各関心事126a、126bはキーワード／タグ（例えば、concern3、concern4）と一緒にリスト132に格納されてもよい（特徴2.1）。 20

【0045】

アスペクト124がアプリケーションに実装を追加できる関心事（例えば、関心事126a）は、アスペクト124がアプリケーションから削除できる関心事（例えば、関心事126b）から区別されるべきである。1つの実施形態では、これを実現するために、関心事126a、126bについて別個のリスト132a、132bを用いる（特徴2.2）。1つのリスト132aはアスペクト124がアプリケーションに追加する関心事126aを識別し、別のリスト132bはアスペクト124がアプリケーションから削除する関心事126bを識別する。アプリケーションへの関心事の削除及び追加の例は、元の合成サービス118がユーザに課金する機能を含むかもしれない場合である。しかしながら、オペレータは、課金の特定のメソッドを有することを望まないかも知れず、それ故、この種の課金を削除するためにアスペクト124（関心事126b）を使用する。さらに、オペレータは別のアスペクト124（関心事126a）を通じて自身の好みの課金方法を合成サービス118に追加してもよい。関心事126a、126bは両方の場合について「charging」と呼ばれないだろうが、例えば、「charging_methodA」、「charging_methodB」と呼ばれるだろう。 30

【0046】

各関心事126a、126bは、それぞれが関心事の機能全体の一部を実装する複数のアスペクトによって実装されてもよい。これらのアスペクトはアスペクト・グループである。これらは共通アスペクト・グループIDと一緒にアスペクト・リポジトリ106に登録される（特徴2.3）。このアスペクト・グループIDを実装する方法は複数存在する。例えば、特定のグループに属するすべてのアスペクト124にグループIDを付加することによる。または、特定のグループに属するすべてのアスペクト124のリストとして別々にグループIDを特定することによる。これらのアプローチのそれぞれは独自の利点及び不都合を有する。 40

【0047】

2. ランタイムでの関心事の処理（特徴3.x、4.x）

前節で導入されたサービス記述122及びアスペクト記述128はサービス108の実行のための環境を設定する。サービス記述122、128はサービス108のランタイム 50

で用いられる（特徴3）。

【0048】

実行開始において、アプリケーションサーバ102、特に合成エンジン110はサービス記述122をサービス・インスタンスのランタイム・データ114にロードする。1つの実施形態では、これを行うために、サービス記述122をサービス・リポジトリ104からランタイム・データ114へコピーする（特徴3. 1）。別の実施形態では、分析機能112はサービス108の分析の結果としてサービス記述122を判定しうる。

【0049】

ランタイム・サービス関心事記述122'はアスペクト124がアプリケーションにウィービングされるごとに更新される（特徴3. 2）。1つの実施形態では、アプリケーションサーバ102、特にウィービング・エンティティ116はこれを行うために、アスペクト関心事記述128に特定される関心事をランタイム・サービス関心事記述122'に追加する（特徴3. 3）。

【0050】

アスペクト124が機能を削除するならば、それぞれの関心事記述は追加されるのではなく、ランタイム・サービス関心事記述122'から削除される必要がある（特徴3. 4）。1つの実施形態では、関心事の削除は、当該関心事が削除されたものとしてマークするのではなく、ランタイム・サービス関心事記述122'において実際に離すことによって実行される（特徴3. 5）。

【0051】

ランタイム中に変更されるランタイム・サービス関心事記述122'内のすべての関心事は、変更の起源としてマークされる。例えば、ウィービングされたアスペクトが変更を引き起こすならば、この特定の関心事はそれぞれのアスペクトIDによってマークされる（特徴3. 6）。

【0052】

所与のジョインポイントについて、関心事を実装する複数のアスペクトが適用可能であってもよい。例えば、課金のためのアスペクトの適用を導くジョインポイントにおける2つの有効なウィービング命令が存在してもよい。二重課金を避けるために、これらのうちの1つだけが実際に用いられるべきである。1つの実施形態では、ウィービング・エンティティ116はウィービングについて1つのアスペクトだけ（例えば、処理された最初のアスペクト）を選択し、同じ関心事のための他のアスペクトはスキップされる。別の実施形態では、どのアスペクトが用いられるかを制御するためにポリシーが定義されうる（特徴4. 3、4. 4）。単一の関心事について複数のアスペクトが用いられるならば、複数のアスペクトIDを関心事に割り当てることによって、ランタイム関心事記述122'にこれが反映されうる（特徴3. 7）。

【0053】

ランタイム・サービス関心事記述122'において、アスペクトのIDはアスペクトが追加した関心事と一緒に格納される。相異なるジョインポイントで同じ関心事のための同じアスペクトがブロックされないことを保証するためにこれが行われる。そうでなければ、アスペクトは自身をブロックしうる。これは、ランタイム・サービス関心事記述122'が最初のジョインポイントで更新されるので発生しうる。その後、更新されたランタイム・サービス関心事記述122'内にこの関心事が既にあるので、他のジョインポイントにおいて、アスペクトの他の部分のウィービングがブロックされる。関心事の実装が複数のアスペクトに亘って意図的に広がるならば、このようなブロックも発生しうる。単一の関心事について複数のアスペクトが用いられるならば、複数のアスペクトIDを関心事に割り当てることによって、ランタイム・サービス関心事記述122'にこれが反映されうる（特徴3. 7）。これにより、関心事に割り当てられたすべてのアスペクトが、同じ関心事に対処する他のジョインポイントで用いられることが可能になる。

【0054】

特定のアスペクトのウィービングを導くポリシーは変わりうる。これが発生したならば

10

20

30

40

50

、ランタイム・サービス関心事記述122'は新たなポリシーを反映するために更新される必要があるだろう。この更新はランタイム・サービス関心事記述122'において関心事に割り当てられたアスペクトIDの削除を暗に示すだろう。これはまた、新たなポリシーに従って選択されたいだろアスペクトの新たなIDで元のアスペクトIDを置き換えることにつながるだろう（特徴3.6）。しかしながら、このメカニズムは、最初に用いられたものとは非互換である残りの実行のためのアスペクトが用いられないことを保証するように、慎重に用いられるべきである。さらに、置換アルゴリズムは互換性問題を検討し考慮に入れるべきである。実際、すべての実行中の合成セッションについてポリシーを有効に維持し、新たなセッションについてのみ新たなポリシーを使用の方がよいかもしれない。いずれの場合でも、例えば合成の再開や古いポリシーの更なる使用が受け入れられない長時間実行中の合成セッションについて、記載されたポリシーの置換が必要になりうる。

10

【0055】

時には、一般に様々なジョインポイントでアプリケーションにウェーブされる複数のアスペクトに関心事の実装が分配される必要がある。このようなアスペクトのグループは単一のグループアスペクトIDと一緒にアスペクト・リポジトリ106に記述される。その後、グループのアスペクトIDは、グループのアスペクトが適用されるとランタイム・サービス関心事記述122'内の関心事にも割り当てられる。同じグループからの異なるアスペクトが別のジョインポイントでウィービングされようとしているならば、これは許可される。なぜなら、このアスペクトは、既に関心事に割り当てられた同じアスペクト・グループIDと一緒にアスペクトが記述されるからである（特徴3.8）。

20

【0056】

3. ウェービング決定に影響する関心事（特徴4.x）

ランタイムにおいて、アスペクト124はアプリケーションにウィービングされ、各ジョインポイントにおいて、ウィービング・エンティティ116はアスペクト124がウィービングされうるかどうかに関する決定を行う必要がある。ウィービング・エンティティ116は、アスペクト124が実装する関心事126a、126bが対象アプリケーションに既に存在するならば、アスペクト124のウィービングをスキップする。この目的のために、アスペクト・リポジトリ106に格納されたアスペクト関心事記述128はランタイム・サービス関心事記述122'に対して比較される。アスペクト関心事126a、126bが既にランタイム・データ114内にあるならば、ウィービングはスキップされてもよい。

30

【0057】

アスペクト124はアプリケーションに亘る複数のジョインポイントでウィービングされてもよい。これは、単に最初のウィービングにおいてランタイム・サービス関心事記述122'が更新されたからといって、更なるウィービングを常に拒絶してはならないことを意味する。更なるウィービングが同じアスペクト124に関連しうるならば、これは許可されるべきである。これを達成するために、アスペクト関心事記述タグをランタイム・サービス能力リストに格納するだけでなく、アスペクトID130が特定の関心事120a、120bと一緒に格納される。同じアスペクト124の更なるウィービングにおいて、ウィービング・エンティティ116はランタイム・データ124にマッチング関心事記述及びアスペクト記述128を発見するが、対立する関心事が同じアスペクトID130を有するアスペクト124によって追加されたならばウィービングは許可される。

40

【0058】

ジョインポイントにウィービングされるべきアスペクト124に関する決定を行う際に、同じ関心事120a、120bについての複数のアスペクト124が適用可能であってもよい。1つの実施形態では、ポリシーは、どのアスペクト124を選択するかに関する規則を提供する。例えば、ポリシーは検討される最初のアスペクト124を特定してもよい。別の関心事120a、120bについて、ジョインポイントについて適用可能な最後のアスペクトが選択されてもよい。ポリシーはまた、複数のアスペクト124が関心事1

50

20 a、120 bについてウィーピングされることを許可し、アスペクト・ウィーピング／実行の順序を判定してもよい。

【0059】

4. その他の実施形態（特徴5）

図1に示されるように、合成に利用可能な構成サービス121 a、121 bと、ウィーピングに利用可能なアスペクト124とは別個のリポジトリ104、106内に登録される。必要ならば、ただ1つのリポジトリが、（構成サービス121 a、121 bを含む）合成サービス118とアスペクト124との両方のために存在しうる。サービス及びアスペクト関心事記述122、128のための具体的なオントロジーの定義はドメイン固有でありうる。

10

【0060】

1つの実施形態では、デフォルトの関心事120 a、120 bのリスト119は構成サービス118について事前に定義され、サービス・リポジトリ104内に格納される。別の実施形態では、合成サービス118が既に対処している関心事120 a、120 bのリスト119は、この合成サービス118をインスタンス化するために必要な構成サービス121 a、121 bに関して合成サービス118を分析する分析機能112によって自動的に生成されてもよい（特徴5）。

【0061】

5. 例示のシナリオ（図2、図3）

サービス合成環境内で、アスペクト124を合成サービス118に適用するためにオンライン・ウィーピングが用いられうる。図2に示される例では、使用されるすべての構成サービス121 a'、121 b'、121 c'、121 d'、121 e'、121 f'のロギングを合成サービスアプリケーション118に追加するアスペクト124が使用される。これは、合成サービス118内の各サービス選択後にロギング・アスペクト124が呼び出されることを意味する。よって、各ジョインポイントにおいて、ウィーピング・エンティティ116は、図3に関連するフローチャートに示されるウィーピング決定ロジックに従って決定を行う。

20

【0062】

図3を参照して、図2に示される例示のサービスを実装するためにウィーピング・エンティティ116によって実行されるウィーピング決定ロジック300を説明するフローチャートを示す。ウィーピング・エンティティ116は、ジョインポイントに到達した（ステップ302）後に、ウィーピング命令に従ってアスペクト124を選択し（ステップ304）、アスペクト関心事記述128を取得する（ステップ306）ように機能する。その後、ウィーピング・エンティティ116は、アスペクト関心事がランタイム・サービス制約のリストにあるかを判定し（ステップ308）、結果がNOであればアスペクト関心事がサービス関心事122'のランタイム・リストに追加され（ステップ310）、アスペクト124がアプリケーションにウィーピングされ（ステップ312）、その後終了する（ステップ314）。アスペクトが実際にランタイム・サービス制約のリストにあることをウィーピング・エンティティ116が判定した（ステップ308）ならば、ランタイム関心事が同じアスペクト124からのものであるかを判定し（ステップ316）、結果がNOであれば終了し（ステップ318）、結果がYESであればアスペクト124をアプリケーションにウィーピングし（ステップ320）、その後終了する（ステップ322）。この例では、サービスはロギングをサポートしない。ロギングはランタイムでアスペクトによって追加される。最初に、ロギング・アスペクトが適用されるジョインポイントに到達した場合に、ウィーピング・エンティティ116は、アプリケーションによって既に対処されている関心事のリストにロギング関心事がないことを判定すると仮定する（ステップ308）。アスペクトは適用され、このサービス・インスタンスについて対処される関心事のリストは更新される（ステップ310、312）。次のジョインポイントで、再びロギング・アスペクトを適用する場合に（ステップ308）、ウィーピング・エンティティ116は、ロギング・アスペクトの関心事が既に対処される関心事のリスト内にあ

30

40

50

ることを見つける。このロギング関心事は同じIDのアスペクトによって追加されたので、ウィービング・エンティティ116はこのアスペクトの使用を再び許可することを決定する(ステップ320)。やはりロギングを適用するだろう別のアスペクトが試されたなら、ウィービング・エンティティ116は2つの独立したアスペクトが同じ関心事に対処することを避けるためにウィービングを拒絶するだろう(ステップ318)。

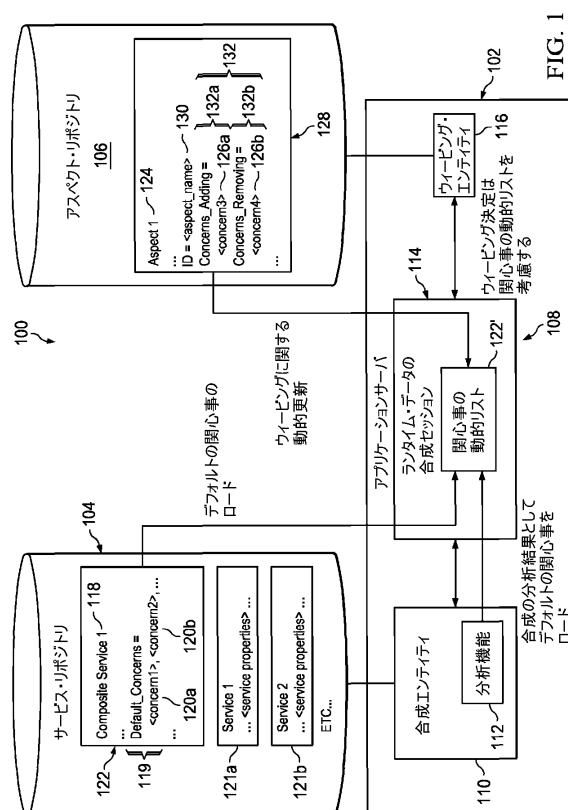
【0063】

上記により、当業者は、本発明は少なくとも先行技術に関する前述の欠点を軽減し、アスペクト指向サービス管理環境全体の中央構成要素を提供することが意図されることを理解するだろう。このような環境では、ウィービング言語により複数のアプリケーション及びサービスに広域に亘るウィービングを定義できる。本発明は、アスペクトが必要なアプリケーションにのみアスペクトがウィービングされることを保証することに役立つ。特に、新たなアプリケーションが環境に追加されるならば、本明細書に記載されたメカニズムはグローバルなウィービング命令を変更することなくアスペクトの自動適用を可能にする。なぜなら、アスペクトがこのアプリケーションにウィービングされるべきかどうか、及びどのアスペクトがウィービングされるべきかをアプリケーションサーバが自動且つ動的に決定できるからである。このようなサービス環境のオペレータは、アプリケーションの機能に関して自身の要件を表現するグローバルなウィービング命令を提供できる。本明細書に記載されるメカニズム及び特徴はサポートを含まないアプリケーションにアスペクト・ウィービングによってこの機能を追加するのも役立つ。この要件に既に準拠しているアプリケーションはウィービングから自動的に除外される。

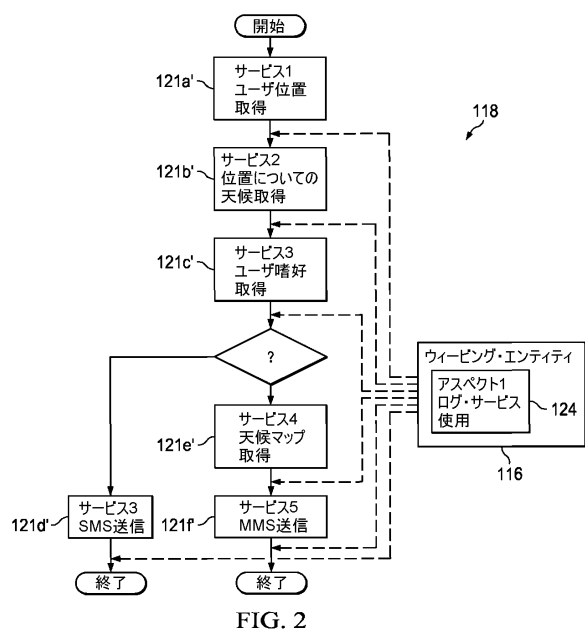
【0064】

本発明の複数の実施形態が添付の図面に説明され、前述の詳細な説明に記載されてきたが、本発明は開示された実施形態に限られず、添付の特許請求の範囲で説明され規定される本発明から逸脱することなく、多数の再構成、変形及び代替が可能であることが理解されるべきである。

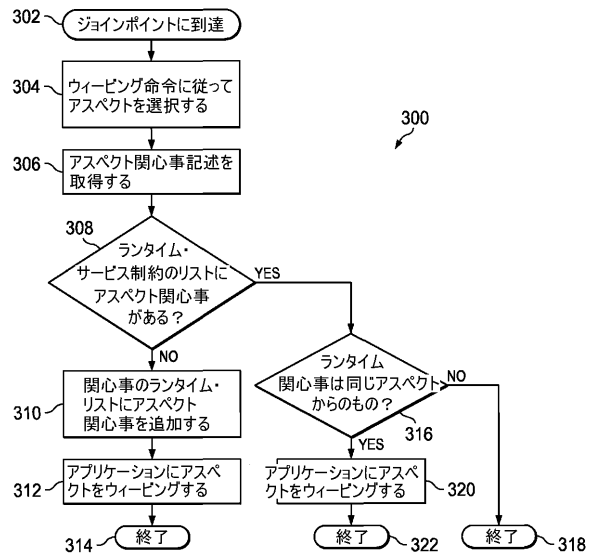
【図1】



【図2】



【図 3】



フロントページの続き

(74)代理人 100161399

弁理士 大戸 隆広

(72)発明者 ニーメラー, ヨルク

スウェーデン国 スンドビュベリ 17241, スコグスバッケン 20

(72)発明者 フライター, オイゲン

ドイツ国 ベルリン 14059, シュパンダウアー ダム 55

(72)発明者 ルヴェンシュテイン, ロマン

ドイツ国 アーヘン 52074, ミースーファンデーデルローエーシュトラッセ 35

(72)発明者 キネ, ラファエル

ベルギー国 リエージュ ビー-4000, ヴィゼーヴォワ, 66

審査官 坂庭 剛史

(56)参考文献 豊田陽一、他2名、コンパイル時に織り込み位置を決定する動的アスペクト指向プログラミング、情報処理学会研究報告、社団法人情報処理学会、2004年 6月18日、第2004巻、第63号、第73から80頁

皆本房幸、第1章 オープンソースJ2EEサーバJBoss入門、WEB+DB PRESS、日本、(株)技術評論社、2003年 9月20日、第16巻、第48から53頁

和田周、追跡!アスペクト指向プログラミング、Java WORLD、日本、(株)IDGジャパン、2004年 4月 1日、第8巻、第4号、第81から91頁

赤堀文隆、他2名、アスペクト指向プログラミングのためのアドバイス適用順序をテストする枠組みの提案、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、2008年11月20日、第108巻、第326号、第55から59頁

中村正、第1章 AspectJ、JAVA PRESS、日本、(株)技術評論社、2003年 2月15日、第28巻、第2から11頁

(58)調査した分野(Int.Cl., DB名)

G06F 9/44