

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6283654号
(P6283654)

(45) 発行日 平成30年2月21日(2018.2.21)

(24) 登録日 平成30年2月2日(2018.2.2)

(51) Int.Cl.

F I

G 0 6 F 8/70 (2018.01)

G 0 6 F 9/06 6 2 0 K

請求項の数 20 (全 26 頁)

(21) 出願番号	特願2015-503457 (P2015-503457)	(73) 特許権者	514244572
(86) (22) 出願日	平成25年3月26日(2013.3.26)		ビーエムシー ソフトウェア、インコーポ レーテッド
(65) 公表番号	特表2015-515684 (P2015-515684A)		アメリカ合衆国 テキサス州 77042
(43) 公表日	平成27年5月28日(2015.5.28)		ヒューストン シティーウエスト ブル ーバード 2101
(86) 国際出願番号	PCT/US2013/033839	(74) 代理人	100097456
(87) 国際公開番号	W02013/148651		弁理士 石川 徹
(87) 国際公開日	平成25年10月3日(2013.10.3)	(72) 発明者	アブヒジト シャルマ
審査請求日	平成28年3月25日(2016.3.25)		インド国 プネ 411001 マハラシ ユトラ 98 コレガオン パーク レー ン 8 ウェリントン メウス ウィング 3 103
(31) 優先権主張番号	13/433,162		
(32) 優先日	平成24年3月28日(2012.3.28)		
(33) 優先権主張国	米国 (US)		

最終頁に続く

(54) 【発明の名称】 アプリケーションをアセンブルするためのブループリント自動アセンブリ

(57) 【特許請求の範囲】

【請求項 1】

ブループリントの自動アセンブリを実行するデータ処理装置であって、

アプリケーションのアプリケーション・ブループリントを自動的にアセンブルするための、ブループリントの自動アセンブリを実行させる要求を受け取るよう構成されるマイクロ・ブループリント・アセンブラであって、該要求が、少なくとも1つの特徴を規定する、前記マイクロ・ブループリント・アセンブラと、

モデルデータを格納するよう構成されるモデルのデータベースであって、該モデルデータが、複数のクラス間のリンクを規定する関係情報を用いて階層構造で配列される該複数のクラスを含む、前記モデルのデータベースと、

複数のマイクロ・ブループリントを格納するよう構成されるマイクロ・ブループリントのデータベースであって、各マイクロ・ブループリントが、機能コンポーネントに対応し、かつウェブ層、アプリケーション層、及びデータベース層のうちの少なくとも1つのメンバーのサービススタックエレメントをそれぞれ含み、各マイクロ・ブループリントには、該モデルのデータベースの該モデルデータから該複数のクラスのうちの1つ以上のクラスの注釈が付けられる、前記マイクロ・ブループリントのデータベースと、

実行時間中に、該要求に従って、該モデルデータ及び該複数のマイクロ・ブループリントに基づいて、該アプリケーション・ブループリントを生成するよう構成されるマイクロ・ブループリント・アセンブラであって、

該アプリケーション・ブループリントを生成するときに、該モデルのデータベースの該

10

20

モデルデータのクラスに、該少なくとも1つの特徴を適用して、該要求の該少なくとも1つの特徴に関連するクラスを取得し、該マイクロ・ブループリントのデータベースを検索して、該取得されたクラスを有する少なくとも1つのマイクロ・ブループリント及び該関係情報によって該取得されたクラスにリンクされた、少なくとも1つのマイクロ・ブループリントを取得するように構成されている、前記マイクロ・ブループリント・アセンブラと、を備える、

前記データ処理装置。

【請求項2】

前記要求がまた、少なくとも1つの制約及び環境も規定する、請求項1記載のデータ処理装置。

【請求項3】

前記少なくとも1つの特徴が、セキュリティ特徴、又は拡張性特徴を規定する非機能的特徴である、請求項1記載のデータ処理装置。

【請求項4】

前記複数のクラスが、前記サービススタックエレメントの異なるレベルを表わす、請求項1記載のデータ処理装置。

【請求項5】

前記複数のクラスが、アプリケーション、デプロイメント、アプリケーションサーバ、プラットフォームランタイム、オペレーティングシステム、及びデータベースサーバのうちの少なくとも1つを表わすコアの抽象クラスを含み、各前記コアの抽象クラスが、前記サービススタックエレメントに関する前記マイクロ・ブループリントに対応するサブクラスを含む、請求項1記載のデータ処理装置。

【請求項6】

前記マイクロ・ブループリント・アセンブラは、前記ウェブ層、前記アプリケーション層、前記データベース層に関するマイクロ・ブループリントをアセンブルするよう構成される、請求項1記載のデータ処理装置。

【請求項7】

前記マイクロ・ブループリント・アセンブラが、

前記アプリケーションの前記ウェブ層、前記アプリケーション層、及び前記データベース層に対応するマイクロ・ブループリントをアセンブルするよう構成されるアプリケーション・ブループリント・アセンブラと、

前記ウェブ層、前記アプリケーション層、及び前記データベース層ごとの前記サービススタックエレメントに対応するマイクロ・ブループリントをアセンブルするよう構成される機能コンポーネントのブループリント・アセンブラと、を備える、請求項1記載のデータ処理装置。

【請求項8】

前記アプリケーション・ブループリント・アセンブラは、前記マイクロ・ブループリントをアセンブルするよう構成され、この前記マイクロ・ブループリントをアセンブルする動作には、

前記要求及び前記複数のマイクロ・ブループリントの必要な能力及び有効な能力に従って、前記マイクロ・ブループリントのデータベースから前記ウェブ層、前記アプリケーション層、及び前記データベース層に対応するマイクロ・ブループリントを取得することが含まれる、請求項7記載のデータ処理装置。

【請求項9】

前記機能コンポーネントのブループリント・アセンブラは、前記マイクロ・ブループリントをアセンブルするよう構成され、この前記マイクロ・ブループリントをアセンブルする動作には、

前記要求及び前記関係情報に従って、前記ウェブ層、前記アプリケーション層、及び前記データベース層ごとの前記サービススタックエレメントに対応するマイクロ・ブループリントを取得することが含まれる、請求項7記載のデータ処理装置。

【請求項 10】

前記アプリケーション・ブループリント・アセンブラは、前記マイクロ・ブループリントをアセンブルするよう構成され、この前記マイクロ・ブループリントをアセンブルする動作には、

ヒューリスティックベースの最良優先探索アルゴリズムを用いて、前記マイクロ・ブループリントのデータベースからマイクロ・ブループリントを取得することが含まれる、請求項7記載のデータ処理装置。

【請求項 11】

前記マイクロ・ブループリント・アセンブラが、前記要求を実現させるための適合性の順番でアプリケーションのブループリントを並べたリストを生成するよう構成される、請求項1記載のデータ処理装置。

10

【請求項 12】

ブループリントの自動アセンブリを実行する方法であって、

アプリケーションのアプリケーション・ブループリントを自動的にアセンブルするための、ブループリントの自動アセンブリを実行させる要求を受け取る工程であって、該要求が、少なくとも1つの特徴を規定する、前記工程と、

モデルのデータベースからモデルデータを取得する工程であって、該モデルデータが、複数のクラス間のリンクを規定する関係情報を用いて階層構造で配列される該複数のクラスを含む、前記工程と、

マイクロ・ブループリントのデータベースからマイクロ・ブループリントを取得する工程であって、各マイクロ・ブループリントが、機能コンポーネントに対応し、かつウェブ層、アプリケーション層、及びデータベース層のうちの少なくとも1つのメンバーのサービススタックエレメントをそれぞれ含み、各マイクロ・ブループリントには、該モデルのデータベースの該モデルデータから該複数のクラスのうちの1つ以上のクラスの注釈が付けられる、前記工程と、

20

実行時間中に、該要求に従って、該モデルデータ及び該マイクロ・ブループリントに基づいて、該アプリケーション・ブループリントを生成する工程であって、該要求が、該モデルのデータベースの該モデルデータからリンクされたクラスを抽出し、その後、該マイクロ・ブループリントのデータベースから該リンクされたクラスを有する該マイクロ・ブループリントを取得することを含む、前記工程、を含む、前記方法。

30

【請求項 13】

前記要求がまた、少なくとも1つの制約及び環境も規定する、請求項12記載の方法。

【請求項 14】

前記少なくとも1つの特徴が、セキュリティ特徴、又は拡張性特徴を規定する非機能的特徴である、請求項12記載の方法。

【請求項 15】

前記複数のクラスが、サービススタックエレメントの異なるレベルを表わす、請求項12記載の方法。

【請求項 16】

前記複数のクラスは、アプリケーション、デプロイメント、アプリケーションサーバ、プラットフォームランタイム、オペレーティングシステム、及びデータベースサーバのうちの少なくとも1つを表わすコアの抽象クラスを含み、各前記コアの抽象クラスが、該サービススタックエレメントに関する前記マイクロ・ブループリントに対応するサブクラスを含む、請求項12記載の方法。

40

【請求項 17】

前記アプリケーションのブループリントを生成する工程には、

前記要求に従って、各サービススタックエレメントに関する、マイクロ・ブループリントをアセンブルすることが含まれる、請求項12記載の方法。

【請求項 18】

前記アプリケーション・ブループリントを生成する工程には、

50

前記アプリケーションの前記ウェブ層、前記アプリケーション層、及び前記データベース層に対応するマイクロ・ブループリントをアセンブルする工程と、

前記ウェブ層、前記アプリケーション層、及び前記データベース層ごとの前記サービススタックエレメントに対応するマイクロ・ブループリントをアセンブルする工程と、が含まれる、請求項12記載の方法。

【請求項19】

実行されると1つ以上のプロセッサが処理を行う命令を格納する非一時的なコンピュータ可読媒体であって、該命令には、

アプリケーションのアプリケーション・ブループリントを自動的にアセンブルするための、ブループリントの自動アセンブリを実行させる要求を受け取る命令であって、該要求が、少なくとも1つの特徴を規定する、前記命令と、

モデルのデータベースからモデルデータを取得する命令であって、該モデルデータが、複数のクラス間のリンクを規定する関係情報を用いて階層構造で配列される該複数のクラスを含む、前記命令と、

マイクロ・ブループリントのデータベースからマイクロ・ブループリントを取得する命令であって、各マイクロ・ブループリントが、機能コンポーネントに対応し、かつウェブ層、アプリケーション層、及びデータベース層のうちの少なくとも1つのメンバーのサービススタックエレメントをそれぞれ含み、各マイクロ・ブループリントには、該モデルのデータベースの該モデルデータから該複数のクラスのうちの1つ以上のクラスの注釈が付けられる、前記命令と、

実行時間中に、該要求に従って、該モデルデータ及び該マイクロ・ブループリントに基づいて、該アプリケーション・ブループリントを生成する命令であって、該要求が、該モデルのデータベースの該モデルデータからリンクされたクラスを抽出し、その後、該マイクロ・ブループリントのデータベースから該リンクされたクラスを有する該マイクロ・ブループリントを取得することを含む、前記命令と、を含む、前記非一時的なコンピュータ可読媒体。

【請求項20】

前記マイクロ・ブループリントが、前記アプリケーションのビルディング・ブロックを表わす、再使用可能なテンプレートである、請求項19記載の非一時的なコンピュータ可読媒体。

【発明の詳細な説明】

【技術分野】

【0001】

(関連出願の相互参照)

本出願は、2012年3月28日出願の「アプリケーションをアセンブルするためのブループリント自動アセンブリ (AUTOMATED BLUEPRINT ASSEMBLY FOR ASSEMBLING AN APPLICATION)」と題する米国特許出願番号第13/433,162号の優先権の主張をし、当該出願を継続する。また当該出願の内容は引用により本明細書中に完全に組み込まれるものとする。

【0002】

(技術分野)

本開示は、一般に、ブループリントの自動アセンブリに関する。

【背景技術】

【0003】

アプリケーション又はサーバを配置するためには、トポロジ (例えば、層の数)、構成、動作、制約、オペレーティングシステム、及びソフトウェアパッケージの供給が必要であるが、これらはアプリケーションの機能ブループリントにより規定することができる。リソースセット及び必要な計算、記憶容量、ネットワークへの割り当てに関して、アプリケーションを提供できるであろうと思われる方法 (数ある中から) を、各機能ブループリントの配置ブループリントが規定する。例えば、「QA」の配置ブループリントは、1つの仮想マシン (VM) で単一のリソースセットを用いて、アプリケーションの3つの層の全て

10

20

30

40

50

をホストすることができ、一方、「生産」の配置ブループリントは、3つの個々のアプリケーション層を3つの異なるリソースセットに分配することができる。

【発明の概要】

【発明が解決しようとする課題】

【0004】

現在、機能ブループリント又は配置ブループリントを生成するためには、複数の異なる従来の方法が存在する。しかし、これらの従来の方法では、例えば、全ての機能コンポーネント及びそれらの各ソフトウェア・スタックをそのままの状態で規定しているアプリケーションのテンプレートなどの一体型設計によるブループリントが提供されている。その結果として、従来のブループリントは、幅広い再使用には向いていない可能性がある。例えば、ソフトウェア・スタック内のソフトウェア・スタックの要素に対して異なる複数の選択肢が存在する可能性がある。現状の一体型設計を用いると、複数の選択肢が存在し得ることで、全ての異なる選択肢のために複数の機能ブループリントを作成することを意味しかねない。再使用可能なサービス、同様の適合基盤 (conformant infrastructure)、ソフトウェア・スタックの要素、及び/又は、ソフトウェアのバージョンを完全に分けて選択することにより、カタログ内で規定され、維持される完全なブループリントが急増してしまう恐れがある。その結果、従来のブループリントを管理できなくなる可能性がある、あるいは、面倒になってしまう。

【課題を解決するための手段】

【0005】

本実施形態により、ブループリントの自動アセンブリを実行するデータ処理装置が提供される。このデータ処理装置は、アプリケーションをアセンブルするための、ブループリントの自動アセンブリを実行させる要求を受け取るよう構成されるマイクロ・ブループリント・アセンブラであって、この要求が、少なくとも1つの特徴を規定する、マイクロ・ブループリント・アセンブラと、モデルデータを格納するよう構成されるモデルのデータベースと、を備える。このモデルデータには、関係情報を用いて階層構造で配列される複数のクラスと、これらの複数のクラスのうちの少なくとも一部に関するクラスプロパティが含まれる。このデータ処理装置は、複数のマイクロ・ブループリントを格納するよう構成されるマイクロ・ブループリントのデータベースをさらに備える。各マイクロ・ブループリントは、スタック要素、又はサービス層の機能コンポーネントに対応し、この機能コンポーネントには、これらの複数のクラスのうちの1つ以上のクラス、並びに、少なくとも1つの必要な能力、及び有効な能力の注釈が付けられる。このマイクロ・ブループリント・アセンブラは、上記の要求に従って、モデルデータ及び複数のマイクロ・ブループリントに基づいて、少なくとも1つのアプリケーションのブループリントを生成するよう構成されている。

【0006】

一実施形態では、上記の要求がまた、少なくとも1つの制約及び環境も規定する。また、少なくとも1つの特徴は非機能的特徴でよい。複数のクラスはスタック要素の異なるレベルを表すことができ、関係情報は複数のクラス間の関係を規定することができる。

【0007】

複数のクラスには、アプリケーション、デプロイメント、アプリケーションサーバ、プラットフォームランタイム、オペレーティングシステム、及びデータベースサーバのうちの少なくとも1つを有するコアの抽象クラスが含まれ、各コアの抽象クラスは、スタック要素に関するマイクロ・ブループリントに対応するサブクラスを含むことができる。

【0008】

このマイクロ・ブループリント・アセンブラは、少なくとも1つのアプリケーションのブループリントを生成するよう構成され、この少なくとも1つのアプリケーションのブループリントを生成する動作には、上記の要求に従って、各サービス層に対して、及び各サ

ービス層内の各スタックエレメントに対して、複数のマイクロ・ブループリントのサブセットをアセンブルすることが含まれ得る。

【0009】

このマイクロ・ブループリント・アセンブラは、アプリケーションのサービス層に対応するマイクロ・ブループリントをアセンブルするよう構成されるアプリケーション・ブループリント・アセンブラと、各サービス層のスタックエレメントに対応するマイクロ・ブループリントをアセンブルするよう構成される機能コンポーネントのブループリント・アセンブラと、を備えることができる。

【0010】

このアプリケーション・ブループリント・アセンブラは、マイクロ・ブループリントをアセンブルするよう構成され、このマイクロ・ブループリントをアセンブルする動作には、要求、並びに、複数のマイクロ・ブループリントの必要な能力、及び有効な能力に従って、マイクロ・ブループリントのデータベースからサービス層に対応するマイクロ・ブループリントを取得することが含まれ得る。

10

【0011】

この機能コンポーネントのブループリント・アセンブラは、マイクロ・ブループリントをアセンブルするよう構成され、このマイクロ・ブループリントをアセンブルする動作には、要求及び関係情報に従って、各サービス層のスタックエレメントに対応するマイクロ・ブループリントを取得することが含まれ得る。

【0012】

20

このアプリケーション・ブループリント・アセンブラは、マイクロ・ブループリントをアセンブルするよう構成され、このマイクロ・ブループリントをアセンブルする動作には、人工知能(AI)の探索アルゴリズムを用いて、マイクロ・ブループリントのデータベースからサービス層に対応するマイクロ・ブループリントを取得することが含まれ得る。

【0013】

このマイクロ・ブループリント・アセンブラは、要求を実現させるための適合性の順番でアプリケーションのブループリントを並べたリストを生成するよう構成され得る。

【0014】

また本実施形態により、ブループリントの自動アセンブリを実行する方法も提供される。この方法には、アプリケーションをアセンブルするための、ブループリントの自動アセンブリを実行させる要求を受け取る工程であって、この要求が、少なくとも1つの特徴を規定する、工程と、上記の要求に従って、モデルデータ及び複数のマイクロ・ブループリントに基づいて、少なくとも1つのアプリケーションのブループリントを生成する工程と、が含まれる。このモデルデータには、関係情報を用いて階層構造で配列される複数のクラスと、これらの複数のクラスのうちの少なくとも一部に関するクラスプロパティとが含まれる。複数のマイクロ・ブループリントの各マイクロ・ブループリントは、スタックエレメント又はサービス層の機能コンポーネントに対応し、この機能コンポーネントには、複数のクラスのうちの1つ以上のクラス、並びに、少なくとも1つの必要な能力及び有効な能力の注釈が付けられる。

30

【0015】

40

一実施形態では、上記の要求はまた、少なくとも1つの制約及び環境も規定する。また、少なくとも1つの特徴は、非機能的特徴でよい。複数のクラスはスタックエレメントの異なるレベルを表すことができ、関係情報は複数のクラス間の関係を規定することができる。

【0016】

複数のクラスには、アプリケーション、デプロイメント、アプリケーションサーバ、プラットフォームランタイム、オペレーティングシステム、及びデータベースサーバのうちの少なくとも1つを有するコアの抽象クラスが含まれ、各コアの抽象クラスは、スタックエレメントに関するマイクロ・ブループリントに対応するサブクラスを含むことができる。

50

【 0 0 1 7 】

上記の少なくとも1つのアプリケーションのブループリントを生成する工程には、上記の要求に従って、各サービス層に対して、及び各サービス層内の各スタックエレメントに対して、複数のマイクロ・ブループリントのサブセットをアセンブルすることが含まれる。上記の少なくとも1つのアプリケーションのブループリントを生成する工程には、アプリケーションのサービス層に対応するマイクロ・ブループリントをアセンブルすることと、各サービス層のスタックエレメントに対応するマイクロ・ブループリントをアセンブルすることと、が含まれる。

【 0 0 1 8 】

また本実施形態により、実行されると1つ以上のプロセッサが処理を行う命令を格納する非一時的なコンピュータ可読媒体も提供される。この命令には、アプリケーションをアセンブルするための、ブループリントの自動アセンブリを実行させる要求を受け取らせる命令であって、上記の要求が、少なくとも1つの特徴を規定する、命令と、上記の要求に従って、モデルデータ及び複数のマイクロ・ブループリントに基づいて、少なくとも1つのアプリケーションのブループリントを生成させる命令と、が含まれる。このモデルデータには、関係情報を用いて階層構造で配列される複数のクラスと、この複数のクラスのうちの少なくとも一部に関するクラスプロパティとが含まれる。複数のマイクロ・ブループリントのうちの各マイクロ・ブループリントは、スタックエレメント又はサービス層の機能コンポーネントに対応し、この機能コンポーネントには、上記の複数のクラスのうちの1つ以上のクラス、並びに、少なくとも1つの必要な能力及び有効な能力の注釈が付けられる。

【図面の簡単な説明】

【 0 0 1 9 】

【図1】図1は、実施形態に従った、ブループリントの自動アセンブリを実行するデータ処理装置を示す図である。

【 0 0 2 0 】

【図2】図2は、実施形態に従った、図1のデータ処理装置の例示的な動作を示すフローチャートである。

【 0 0 2 1 】

【図3】図3は、実施形態に従った、図1のモデルのデータベースのモデルデータを示す図である。

【 0 0 2 2 】

【図4】図4は、実施形態に従った、ソフトウェア・スタックのエレメント又はサービスを表す機能コンポーネントの注釈を付けるための抽象モデルを示す図である。

【 0 0 2 3 】

【図5】図5は、実施形態に従った、モデルデータのクラスと、ソフトウェア・スタックのエレメントに対応するマイクロ・ブループリントとの間の割り当てを示す図である。

【 0 0 2 4 】

【図6】図6は、実施形態に従った、複数のサービス層に対応する、例示的なマイクロ・ブループリント、及びそれらのクラスの注釈を示す図である。

【 0 0 2 5 】

【図7】図7は、実施形態に従った、複数のサービス層に関して、マイクロ・ブループリントをアセンブルするための工程を示すフローチャートである。

【 0 0 2 6 】

【図8】図8は、実施形態に従った、ソフトウェア・スタックのエレメントに対応するマイクロ・ブループリントをアセンブルするための処理を示すフローチャートである。

【発明を実施するための形態】

【 0 0 2 7 】

図1は、実施形態に従った、ブループリントの自動アセンブリを実行するデータ処理装置100の図である。このデータ処理装置100は、マイクロ・ブループリント・アセンブラ11

10

20

30

40

50

0、モデルのデータベース135、及びマイクロ・ブループリントのデータベース130を備えることができる。モデルのデータベース135はモデルデータを格納し、マイクロ・ブループリントのデータベース130がモデルデータからの情報の注釈を付けたマイクロ・ブループリントを格納する。

【0028】

マイクロ・ブループリントは、ソフトウェア・スタックの要素、及び/又は、サービス層を含むことができる。例えば、各マイクロ・ブループリントは、ソフトウェア・スタックの要素（例えば、Java Oracle JRE 1.6）、又はサービス層の機能コンポーネント（例えば、ウェブ層、アプリケーション層、又はデータベース層）に対応可能である。サービス層用のマイクロ・ブループリントとは、例えば、ウェブ層、アプリケーション層、及びデータベース層など、アプリケーションのより汎用的な機能コンポーネントのことを指すことができる。ソフトウェア・スタックの要素用のマイクロ・ブループリントとは、各サービス層のより専門的な機能コンポーネントのことを指すことができる。例えば、ソフトウェア・スタックの要素に対して、マイクロ・ブループリントは、アプリケーションの各サービス層用のソフトウェア・スタックの要素を備えることができる。一実施形態では、このマイクロ・ブループリントは、例えば、Java実行時環境（JRE）又はアプリケーションサーバのOracle WebLogic 11.5gなどの、ソフトウェア・スタックの要素に関する、且つ「PetStore App- Tier Services」などのサービス層に関する、再使用及び構成可能なブループリントでよい。本実施形態に従うと、これらのマイクロ・ブループリントを一体に組み合わせて、完全なアプリケーションのブループリント、又は機能コンポーネントのブループリントをアセンブルすることが可能である。

【0029】

一般に、ブループリントとは、再使用が可能なアプリケーションの基本的なテンプレートを意味し、このブループリントにより、異なるアプリケーションの図を表すことが可能となる。これらのブループリントは、通常、リポジトリに格納され、例えば、アプリケーションの開発などの種々の目的のために、ユーザが選択するためのカタログの一部として利用可能である。企業及びアプリケーションの設計者は、ミドルウェアの管理者、データベースの管理者、及びオペレーティングシステムの管理者と共同し、例えば、企業のアーキテクチャの標準、基準アーキテクチャ、セキュリティ標準、安定版、標準準拠、及び/又は、性能特性に基づいて、これらのブループリントを規定することができる。すなわち、ブループリントとは、アーキテクチャ図、及び開発図などのアプリケーション又はサービスの異なる局面、或いは図の宣言的仕様記述であり得る。クラウドサービス、及びその他のユースケースに関連する種々のキーアプリケーションを有効にするためにこれらのブループリントが使用される。あるアプリケーションでは、ユーザがブループリントを用いて、クラウド環境内にアプリケーションを供給することが可能である。一般に、ブループリントには、機能ブループリントと配置ブループリントの二種類のブループリントがある。機能ブループリントは、その層（機能コンポーネントとも呼ばれる）に関する、アプリケーション/サービスのアーキテクチャ図、すなわち構造、及びそれらのコンポーネント間の接続を規定することができる。また、この機能ブループリントは、ソフトウェア・スタックの要素、及び各機能コンポーネント内の関連するアーチファクト（スタートアップスクリプト、インストールスクリプト）を規定することもできる。配置ブループリントは、その種々の機能コンポーネントを配置するためのリソース要件（例えば、計算、記憶容量、及びネットワーク）に関して、アプリケーションに関する配置図、すなわち計画を規定することができる。アプリケーションの機能ブループリントには、複数の配置ブループリントが適合可能である。これらの複数の配置ブループリントには、QAの配置ブループリントが含まれ得、このQAの配置ブループリントでは、特定のCPU及びメモリリソースを有する単一の仮想マシン（VM）上に3つの全ての機能コンポーネント（例えば、ウェブ層、アプリケーション層、及びデータベース層）が配置される。一方、生産の配置ブループリントでは、3つの機能コンポーネントはそれぞれ、3つの個々のVMに配置され得る。

これらの複数の配置ブループリントには、QAの配置ブループリントと生産の配置ブループリント以外にも、数多くの異なる種類のブループリントが含まれ得る。

【0030】

下記にさらに詳しく説明するが、データ処理装置100は、ブループリントを構成可能で再使用可能なマイクロ・ブループリントに分割する。そして、これらのマイクロ・ブループリントに、モデルのデータベース135内のモデルデータからの1つ以上のクラスの注釈が付けられる。基本的には、これらの注釈を付けたマイクロ・ブループリントにより、サービス又はソフトウェア・スタックのマイクロ・ブループリントは、1つ以上のモデルデータのクラスに割り当てられ、そして、このモデルデータには、クラス間の関係を規定する関係情報も含まれる。注釈付のマイクロ・ブループリント及びモデルデータを用いることにより、このデータ処理装置100は、完全に構成されたブループリントを、マイクロ・ブループリントから、その場で自動的にアセンブルし、これにより、アプリケーションの供給、すなわち開発が可能となる。すなわち、このデータ処理装置100は、モデル駆動型のアプローチを行って、マイクロ・ブループリントなどの小さなビルディング・ブロックから、完全なブループリントを柔軟で動的に、且つ自動で構成する。上記に示すとおり、この完全なアプリケーションのブループリントには、機能ブループリント、又は配置ブループリントが含まれ得る。この完全なブループリントを用いて、そのコンポーネント又は層と、それらの各ソフトウェア・スタックの要素とを含む完全なアプリケーションを作り出すことができる。

10

【0031】

一実施形態に従うと、このマイクロ・ブループリント・アセンブラ110は、アプリケーションをアセンブルするための、ブループリントの自動アセンブリを実行させる要求を受け取ることができる。この要求により、アプリケーションをアセンブルするための特徴、制約、又は1つ以上の環境を指定することができる。この要求により、アプリケーションをアセンブルする際の、機能的局面（例えば、特徴、サービス）、及び/又は、非機能的局面（例えば、セキュリティ、拡張性）を指定することができる。この要求は、ユーザの要求事項によっては、かなり高いレベル、すなわち粒度の細かいものになり得る。例えば、要求により、ソフトウェア・スタックに関わらず、特定の種類のアプリケーション（例えば、PetStoreのアプリケーション）の必要性を指定することができる。それに応じて、マイクロ・ブループリント・アセンブラ110は、ネットベースのPetStoreのブループリント、又はTomcatベースのPetStoreのブループリントを生成することができる。別の例では、この要求は、J2EEに適合したPetStoreのブループリントを指定することができる。それに応じて、マイクロ・ブループリント・アセンブラ110は、J2EEに適合したブループリントを生成することができる。下記により詳しく説明するとおり、一実施形態では、モデルデータのクラスには特徴が対応可能であり、そのクラスのクラスプロパティには制約及び環境の局面が対応可能である。

20

30

【0032】

一般に、マイクロ・ブループリント・アセンブラ110は、要求に応じて、モデルのデータベース135のモデルデータから、モデルクラス及びそのクラスプロパティのクエリー検索を行うことができる。このモデルデータは、意味的に豊かな拡張性のあるモデルと考えることができ、このモデルデータが、例えば、その領域内のキーのクラスの形態をとる特定分野の知識、クラス階層、クラスプロパティ、及びクラス間の関係、許容値、及び/又は、濃度などのプロパティ、又は関係に対する側面（facets）又は制約を取り込む。このモデルデータは、それ以外にも、クラス及びクラスプロパティに関連する1つ以上の規則も取り込む。本実施形態では、このモデルデータは、例えば、普及しているアプリケーション、コンポーネント、及びソフトウェア・要素、標準アプリケーションサービス、サーバ、及び/又は、ミドルウェアに関連する特定分野の知識を取り込むことができる。キーとなる抽象クラスは、例えば、オペレーティングシステム、プラットフォームランタイム、アプリケーションサーバ、及び/又は、デプロイメントでよい。一実施形態では、このモデルデータには、関係情報を用いて階層構造で配列される複数のクラスが含まれ

40

50

得る。さらに、このモデルデータには、クラスのうちの1つ以上に関するクラスプロパティが含まれ得る。基本的に、これらの複数のクラスがスタックエレメントの異なるレベルを表すことができ、関係情報が複数のクラス間の関係を規定することができる。このモデルデータに関しては、図3を参照して、さらに詳しく説明する。

【0033】

マイクロ・ブループリントのデータベース130に格納されたマイクロ・ブループリントに、モデルデータからの関係のあるキークラスの注釈を付けることで、自動の構成を可能にする。例えば、Oracle Java JREのマイクロ・ブループリントに、モデルからの「OracleJRE」のクラスの注釈を付けることができる。この局面については、図5を参照して、さらに詳しく説明する。つまり、特定の種類のアプリケーションの特徴を規定する要求をマイクロ・ブループリント・アセンブラ110が受け取ると、このマイクロ・ブループリント・アセンブラ110は、モデルのデータベース135内のモデルデータから適切なクラスをクエリー検索し、次いで、マイクロ・ブループリントのデータベース130から、この取得した1つ以上のクラスを有するマイクロ・ブループリントをクエリー検索することができる。さらに、このマイクロ・ブループリントに能力、及び必要な能力の注釈を付けることができるが、これに関しては、図4及び6を参照して、さらに詳しく説明する。

【0034】

このマイクロ・ブループリント・アセンブラ110は、モデルデータ、及び注釈付の再利用可能なマイクロ・ブループリントを活用することができる柔軟性を駆使して、完全なアプリケーションのブループリントのアセンブリを自動化することが可能である。そして、このアプリケーションのブループリントを用いて、そのサービス層と、各サービス層のソフトウェア・スタックのエレメントを含む、完全なアプリケーションを作り出すことができる。すなわち、意味的に豊かな抽象モデル、及び注釈付のマイクロ・ブループリントにより、アプリケーションの要求に基づいて、実行時間中に構成される、完全なアプリケーションのブループリントの自動構成が可能となる。

【0035】

マイクロ・ブループリント・アセンブラ110は、アプリケーション・ブループリント・アセンブラ115と、機能コンポーネントのブループリント・アセンブラ120とを備えることができる。すなわち、ある実施形態では、処理を2つの部分に分割可能であり、この2つの部分の一方はアプリケーション・ブループリント・アセンブラ115が実行する部分であり、もう一方は機能コンポーネントのブループリント・アセンブラ120が実行する部分である。このアプリケーション・ブループリント・アセンブラ115は、マイクロ・ブループリントのデータベース130からのマイクロ・ブループリントをアセンブルすることができ、これらのマイクロ・ブループリントは、サービス層（例えば、ウェブ層、アプリケーション層、及びデータベース層）に対応する。このサービス層用のマイクロ・ブループリントのアセンブリに関しては、図7を参照して、さらに詳しく説明する。レベルが深くなるごとに、機能コンポーネントのブループリント・アセンブラ120は、各サービス層のスタックエレメントに対応するマイクロ・ブループリントのデータベース130から、マイクロ・ブループリントをアセンブルすることができる。このサービス層ごとのスタックエレメントに関するマイクロ・ブループリントのアセンブリに関しては、図8を参照して説明する。また、マイクロ・ブループリントのデータベース130内に比較的大きな数のマイクロ・ブループリントが含まれる場合、アプリケーションのコンポーネント・アセンブラ115は、人工知能（AI）の探索アルゴリズムを用いて、適切なマイクロ・ブループリントを見つけることができる。

【0036】

図2は、図1のデータ処理装置100の動作の例を示すフローチャートである。図2は、連続して順序付けられた動作の一覧として示されているが、これらの動作のうちのいくつか、又は全ては、異なる順序で、又は平行して、或いは繰り返して行うことができる、或いは、時間を重ねて行うことができることは理解されよう。

【0037】

モデルのデータベース内にモデルデータを格納することができる(202)。本実施形態に従うと、モデルのデータベース135にモデルデータを格納することができる。このモデルデータは、例えば、普及しているアプリケーション、コンポーネント、及びソフトウェア・エレメント、標準アプリケーションサービス、サーバ、及び/又は、ミドルウェアに関連する特定分野の知識を取り込むことができる。キーとなる抽象クラスは、例えば、オペレーティングシステム、プラットフォームランタイム、アプリケーションサーバ、及び/又は、デプロイメントでよい。ある実施形態では、このモデルデータは、関係情報を用いて階層構造で配列された、複数のクラスを含むことができる。さらに、このモデルデータは、これらのクラスのうちの1つ以上に関するクラスプロパティを含むことができる。基本的に、複数のクラスは、スタックエレメントの異なるレベルを表すことができ、関係情報は複数のクラス間の関係を規定することができる。

10

【0038】

マイクロ・ブループリントのデータベース内に複数の注釈付のマイクロ・ブループリントを格納することができる(204)。例えば、マイクロ・ブループリントのデータベース130は、複数の注釈付のマイクロ・ブループリントを格納可能である。各マイクロ・ブループリントは、スタックエレメント又はサービス層の機能コンポーネントに対応可能であり、この機能コンポーネントには、複数のクラスのうちの1つ以上のクラス、並びに、少なくとも1つの必要な能力、及び有効な能力の注釈が付けられる。

【0039】

アプリケーションをアセンブルするための、ブループリントの自動アセンブリを実行させる要求を受け取ることができる(206)。例えば、マイクロ・ブループリント・アセンブラ110は、アプリケーションをアセンブルするための、ブループリントの自動アセンブリを実行させる要求を受け取ることができる。この要求は、少なくとも1つの特徴(機能的な、又は非機能的な)を指定することができる。随意的に、この要求は、アプリケーションをアセンブルするための、1つ以上の制約又は環境を指定することもできる。すなわち、この要求は、機能的局面(例えば、特徴、サービス)、及び/又は、非機能的局面(例えば、セキュリティ、拡張性)を指定することができる。ユーザの要求事項によっては、この要求が、かなり高いレベル、すなわち粒度の細かいものになる可能性がある。

20

【0040】

要求に従って、モデルデータ、及び複数のマイクロ・ブループリントに基づいて、少なくとも1つの完全なアプリケーションのブループリントを生成することができる(208)。例えば、マイクロ・ブループリント・アセンブラ110は、要求に応じて、モデルデータ、及び複数のマイクロ・ブループリントに基づいて、少なくとも1つの特徴、制約、又は環境を有するアプリケーションのブループリントのうちの1つ以上を生成することができる。例えば、このマイクロ・ブループリント・アセンブラ 110は、この要求(例えば、特徴、制約、及び/又は環境)に従って、マイクロ・ブループリントのデータベース130から、各サービス層に対して、及び各サービス層内のスタックエレメントに対して、複数のマイクロ・ブループリントのサブセットをアセンブルすることができる。まず、このアプリケーション・ブループリント・アセンブラ115は、アプリケーションのサービス層に応じたマイクロ・ブループリントをアセンブルすることができる。ある実施形態では、アプリケーション・ブループリント・アセンブラ115は、要求(例えば、特徴、制約、及び/又は環境)に従って、サービス層に対応するマイクロ・ブループリントを取得することができ、さらにマイクロ・ブループリントのデータベース130内の複数のマイクロ・ブループリントの必要な能力、及び有効な能力も取得することができる。さらに、マイクロ・ブループリントのデータベース130内に格納された複数のマイクロ・ブループリントが比較的大きい場合、このアプリケーション・ブループリント・アセンブラ115は、マイクロ・ブループリントのデータベース130内のマイクロ・ブループリントを取得するために、AI検索のアルゴリズムを用いることができる。その後、機能コンポーネントのブループリント・アセンブラ120は、各サービス層に対するスタックエレメントに対応するマイクロ・ブループリントをアセンブルすることができる。例えば、この機能コンポーネントのブルー

30

40

50

プリント・アセンブラ120は、この要求、及びモデルのデータベース135内のモデルデータの関係情報に従って、各サービス層に対するスタックエレメントに対応するマイクロ・ブループリントを取得することができる。

【0041】

図3は、実施形態に従った、図1モデルのデータベース135のモデルデータ300を示す図である。このモデルデータは、意味的に豊かなモデルを含むことができ、この意味的に豊かなモデルには、それらのアプリケーションのプロビジョニングに対する特定の視点を有する、一般的なアプリケーションのブループリントに関連する、抽象化された特定分野の知識が取り込まれる。モデルのデータベース135は、特定分野の専門家（例えば、設計者、及び/又は、管理者）により、作成され維持され得る。上記に示したとおり、このモデルデータは、例えば、普及しているアプリケーション、コンポーネント、ソフトウェア・エレメント、ミドルウェア、標準アプリケーションサービス、及び/又は、アプリケーションサーバに関連する特定分野の知識を取り込むことができる。

10

【0042】

このモデルデータ300は意味的に豊かな拡張性のあるモデルと考えることができ、この意味的に豊かな拡張性のあるモデルが、例えば、その領域のキークラスの形態の特定分野の知識、クラス階層、クラスプロパティ及びクラス間の関係、許容値、及び/又は、濃度などのプロパティ又は関係に関する側面又は制約を取り込む。それ以外に、このモデルデータはまた、クラス及びクラスプロパティに関連する1つ以上の規則も取り込む。図3を参照すると、このモデルデータ300は、関係情報を用いて階層構造で配列される複数のクラスを含むことができる。図3に示すとおり、複数のクラスはスタックエレメントの異なるレベルを表すことができ、関係情報（例えば、「～である（is a）」、「～を必要とする（needs）」）は、複数のクラス間の関係を規定することができるが、これに関しては下記にさらに詳しく説明する。

20

【0043】

例えば、モデルデータ300は、抽象メインオブジェクト405（例えば、AbstractDCObject）、及び複数のコアのクラス410を含むことができ、これらが一般的なアプリケーションのスタックエレメントと関連する。これらの複数のコアのクラス410は、抽象メインオブジェクト405のサブクラスと考えることができる。ある例では、これらのコアのクラスには、データベースサーバ410-1、オペレーティングシステム410-2、プラットフォームランタイム410-3、アプリケーションサーバ410-4、デプロイメント410-5、及び/又は、アプリケーション410-6が含まれ得る。このアプリケーション410-6は、PetStoreアプリケーション、又はその他の全ての種類のアプリケーションなどのアプリケーションを表すことができる。アプリケーション410-6は、複数のデプロイメントを含むことができ、これらはデプロイメント410-5として表される。このデプロイメント410-5は、パッケージ/アーカイブなどの配置可能なアーチファクト、例えば、ウェブ・アーカイブ（WAR）、エンタープライズ・アーカイブ（EAR）、及び/又は、ダイナミック・ライブ・ライブラリ（DLL）を表すことができ、これらをアプリケーションサーバ内に配置して、アプリケーションのモジュールをインスタンス化する。このデプロイメントのサブクラスは、WAR、EAR、及びDLLでよい。

30

40

【0044】

アプリケーションサーバ410-4（例えば、AppServer）は、その中に配置可能なパッケージを配置して、アプリケーションのモジュールをインスタンス化するアプリケーションサーバを表すことができる。J2EEServer及びIIS（インターネット・インフォメーション・サービス）は、AppServerの2つのサブクラスを表すことができ、さらにその下の階層のWebLogic及びWebSphereは、J2EEServerのサブクラスである。アプリケーションサーバは、実行環境用のプラットフォームランタイムを必要とする。例えば、J2EEServerは、Java JREのプラットフォームランタイムを必要とする。

【0045】

プラットフォームランタイム410-3（例えば、PlatformRuntime）は、その中でアプリケ

50

ーションサーバのようなプログラムが実行されるJava仮想マシン（JVM）などの実行ルーチンの環境を表すことができる。JRE及びnetドメイン（.NET）は、PlatformRuntime410-3のサブクラスであり、さらに、IBM JRE及びOracle JREは、JREのサブクラスである。図3に示すとおり、プラットフォームランタイムは、実行するためのオペレーティングシステムを必要とし得る。

【 0 0 4 6 】

オペレーティングシステム410-2（例えば、OperatingSystem）は、その上でプラットフォームランタイムが実行されるオペレーティングシステムを表すことができる。Unix及びWindowsは、OperatingSystem 410-2のサブクラスであり、さらに、AIX及びLinuxはUnixのサブクラスである。データベースサーバ410-1（例えば、DatabaseServer）は、Oracle又はMSSQLなどのデータベースを表すことができ、このデータベースは、今度は、オペレーティングシステム410-2を必要とする。

【 0 0 4 7 】

クラスのうちの1つ以上は、クラスプロパティを含むことができ、これらのクラスプロパティが、アプリケーションの要求で、ソフトウェア・スタックの要素を選別するための制約として用いられる可能性のあるクラスの様々なプロパティをさらに記述する。すなわち、このモデルデータ300には、これらのクラスの1つ以上に関するクラスプロパティが含まれる。例えば、オペレーティングシステム410-2のクラスプロパティには、32ビットバージョン又は64ビットバージョンなどの異なるバージョンが含まれ得る。つまり、64ビットとなるアプリケーション層のOSバージョンが所望されていることをアプリケーションの要求で指定することができる。さらに、WebSphereのクラスは、WebSphereバージョン5、又はWebSphereバージョン6.1などの異なるWebSphereバージョンを含むことができる。つまり、64ビットとなる最小のアプリケーション層のOSバージョンを有するWebSphereのアプリケーションサーバが所望されていることをアプリケーションの要求で指定することができる。さらに、JREのクラスは、JREのクラスの様々なバージョンを含むことができる。例えば、このJREクラスは、JREバージョン1.5又はJREバージョン1.6を含むことができる。

【 0 0 4 8 】

さらに、このモデルデータ300は関係情報を含むことができる。例えば、このモデルデータは、専門分野モデル内のクラス間の複数のキーの関係を含むことができる。ある例では、このキーの関係には、必要関係が含まれ得る。この必要関係は、AbstractDCObject405上で規定することができ、ソフトウェア・スタックから見たクラス間の従属関係を表すことができる。図3に示されたとおりに説明すると、デプロイメント（例えば、WAR）はAppServer（例えば、WebLogic）を必要とし、AppServerはPlatformRuntime（例えば、OracleJRE）を必要とし、PlatformRuntimeはOperatingSystem（例えば、Linux）を必要とする。

【 0 0 4 9 】

さらに、AppServer及びPlatformRuntimeのサブクラスでは、この必要関係をさらに限定させる。J2EEベースのアプリケーションサーバは、netドメイン（.NET）では、動作することができず、Java実行時環境（JRE）だけで動作することができるため、J2EEServer（AppServerである）は、PlatformRuntimeへの必要関係を、JREを必要とするJRE（すなわちPlatformRuntimeのサブクラス）への関係に限定する。WebSphereに関してさらに別の限定が規定される。WebSphereはIBMの実行時環境（JRE）のみで動作するため、このWebSphereは、IBM JREを必要とする関係を介して、JREのサブクラス（すなわち、IBM JRE）へリンクする。これらを必要関係に対する側面又は制約と考えることができる。

【 0 0 5 0 】

個々のソフトウェア・要素のマイクロ・ブループリントから、好適なソフトウェア・スタックを自動でビルドする処理において、これらの必要関係及びその限定又は制約が利用される。ソフトウェア・スタックから見た、抽象ソフトウェア・要素に関するクラス間の従属関係を、この必要関係により表すことができるため、このようなことが

10

20

30

40

50

可能である。

【0051】

その上、このモデルデータ300は、アプリケーションの非機能的局面を表すことができる。例えば、この非機能的局面（例えば、セキュリティ、拡張性、高可用性）は、専門分野モデル内のクラスとして表すことができ、ソフトウェアパッケージ又は機能コンポーネントのブループリントに注釈を付けるために用いることができる。これにより、機能的局面に加えて、非機能的局面をアプリケーションの要求に含ませることができる。例えば、クラスター化されたWebLogicアプリケーションサーバのマイクロ・ブループリントに、「高可用性」などのコンセプトを表す専門分野からのクラスの注釈を付けることができる。アプリケーションの要求により、高可用性の必要性が示されると、普通のWebLogicアプリケーションサーバのマイクロ・ブループリントよりも優先して、この注釈付のWebLogicアプリケーションサーバのマイクロ・ブループリントが選択される。

10

【0052】

基本的に、これらのマイクロ・ブループリントに関連するクラス（それらのプロパティ及び関係を含む）を用いて、複数のブループリントを動的にアセンブルして、機能コンポーネントごとに、ソフトウェア・スタック用の完全なブループリントを完成させ、これにより、アプリケーション用の完全なブループリントを供給する。

【0053】

このような複数のマイクロ・ブループリントから、完全なアプリケーションのブループリントを、動的に、且つ、自動でアセンブルするためには、上記に説明した、モデルデータ300に属するクラスに関して、それらの機能を記述する仕組み、例えば、ソフトウェアパッケージ又はサービスのマイクロ・ブループリントの機能を記述する仕組みが必要であり得る。例えば、マイクロ・ブループリントには、モデルのデータベース135のモデルデータ300からの情報の注釈が付けられる。具体的には、マイクロ・ブループリントには、1つ以上のクラスの注釈を付けることができる。これに加えて、各マイクロ・ブループリントに能力情報の注釈も付けられるが、これに関しては、図4を参照して、下記にさらに詳しく説明する。

20

【0054】

図4は、実施形態に従った、ソフトウェア・スタックの要素又はサービス層を表す機能コンポーネントの注釈を付けるための抽象モデルを示す図である。例えば、上記に示すとおり、各マイクロ・ブループリントは機能コンポーネント510に対応することができる。機能コンポーネントは、サービス層（例えば、ウェブ層、アプリケーション層、及びデータベース層）、又はソフトウェア・スタックの要素に対応することができる。つまり、機能コンポーネント510は、ソフトウェア・スタックの要素の機能、又はサービス層の機能に対応することができる。機能コンポーネント510は、複数の必要な能力、及び複数の有効な能力を備えることができる。必要な能力はそれぞれ、異なる機能的特徴515に対応することができる。この機能的特徴515には、モデルデータ300からの1つ以上のクラス520の注釈が付けられる。また、有効な能力はそれぞれ、異なる機能的特徴525に対応し、この機能的特徴525には、モデルデータ300からの1つ以上のクラス520の注釈が付けられる。すなわち、抽象モデルは、複数の有効な能力を有する機能コンポーネント510を含むことができ、これにより、各機能コンポーネントの能力を表すことができる。さらに、機能コンポーネント510は、複数の必要な能力を有することができる。これにより、各機能コンポーネントの能力を表すことができる。能力を機能的特徴のクラスの集まりと考えることができる。機能的特徴のクラスは属性タイプを含むことができ、この属性タイプとはモデルデータ300からのクラス520のことを指す。例えば、Oracle Java JREのマイクロ・ブループリントは、モデルデータ300からの「OracleJRE」のクラスの注釈を付けた機能コンポーネントを有する。

30

40

【0055】

図5は、実施形態に従った、モデルデータのクラス300と、ソフトウェア・スタックの要素に対応するマイクロ・ブループリントとの間の割り当てを示す図である。例えば

50

、図5に示すとおり、個々のマイクロ・ブループリント501には、モデルデータ300からの情報の注釈が付けられる。例えば、マイクロ・ブループリント501には、PetStore EARのマイクロ・ブループリント501-1、Oracle WebLogic 11.5のアプリケーションサーバのマイクロ・ブループリント 501-2、Java OracleのJRE 1.6のマイクロ・ブループリント503-3、及びUbuntu Linux 10.04のマイクロ・ブループリント501-4が含まれ得る。これらの種類のマイクロ・ブループリントは説明のためにのみ示されたもので、本実施形態には、全ての種類のソフトウェア・スタックの要素が含まれる。マイクロ・ブループリント501には、モデルデータ300からの関連クラスの注釈を付けることができる（例えば、能力<FunctionalFeature>の種類は、関連クラスを指すことができる）。例えば、Oracle WebLogic 11.5のアプリケーションサーバのマイクロ・ブループリント501-2には、WebLogic
10
のクラスの注釈を付けることができる（J2EEServerのサブクラス-> AppServer -> AbstractDCObject）。また、これらのマイクロ・ブループリントには、このクラス階層内の一般的なクラス、又は限定したクラス、及び、必要に応じて、もちろん複数のクラスの注釈を付けることが可能である。

【 0 0 5 6 】

図6は、実施形態に従った、サービス層に対応する、マイクロ・ブループリント、及びそれらのクラスの注釈を示す図である。図6には、PetStoreのアプリケーションの例が示されているが、本実施形態には、あらゆる数のサービス層を有する全ての種類のアプリケーションが含まれる。図6には、ウェブ層、アプリケーション層、及びデータベース層に関する、3つの異なるマイクロ・ブループリントが示されている。各マイクロ・ブルー
20
プリントは機能コンポーネント510に対応し、機能的特徴515、及び必要とされる機能的特徴525を含む。さらに、機能的特徴515、及び必要な機能的特徴525には、モデルデータ300からのクラス情報の注釈が付けられる。例えば、PetStoreのウェブ層は、「PetStoreWebApp」の能力を有し、機能するためには、「Inventory」及び「ShoppingCart」の能力を必要とする。PetStoreのアプリケーション層は、「商品目録」及び「ShoppingCart」の能力を有し、「DatabaseServer」の能力を必要とする。

【 0 0 5 7 】

ある実施形態では、アプリケーション・ブループリント・アセンブラ115は、必要とされる能力と有効な能力との一致に基づいて、サービス層用のマイクロ・ブループリントをアセンブルすることができ、これにより、PetStoreのウェブ層、PetStoreのアプリケーション
30
層、及びPetStoreのデータベース層のマイクロ・ブループリントを自動的にアセンブルして、完全なPetStore アプリケーションのブループリントを完成させることができる。PetStoreのアプリケーション層などの機能コンポーネント内で、レベルが1つ深くなると、機能コンポーネントのブループリント・アセンブラ120は、データベース130内の注釈付のマイクロ・ブループリントから、完全なソフトウェア・スタック用のマイクロ・ブループリントをアセンブルすることが可能であるが、これに関しては、下記にさらに詳しく説明する。

【 0 0 5 8 】

上記に示したとおり、マイクロ・ブループリント・アセンブラ110は、完全なアプリケーションのブループリントを生成させる要求を受け取ることができる。例えば、このアプリケーションの要求は、少なくとも1つの特徴、及び種々の制約又は環境を指定することができる。アプリケーションの要求を処理することで、動的に構成され、適合性の順番で順位付けられた、アプリケーションのブループリントのリストが出力される。ある実施形態では、このアプリケーションの要求は、例えば、所望される機能的局面（例えば、必要な特徴、サービス）、及び/又は、モデルデータ300からのクラスとして表される非機能的局面（例えば、セキュリティ、拡張性、高可用性）などのある特定の局面を指定することができる。例えば、高可用性を有するPetStoreのアプリケーションに対する要求は、それに応じて、クラスター化されたWebLogic（又はTomcat）のアプリケーションサーバを含む、アプリケーション層の機能コンポーネントのソフトウェア・スタックを用いて、PetStore
40
のアプリケーション用に構成されるブループリントを期待する。それに加えて、その
50

ブループリントは、アプリケーションサーバのインスタンスへのHTTPの要求の負荷を分散させるための、ソフトウェアのロードバランサ用の機能コンポーネントも含むことができる。

【0059】

また、この要求は1つ以上の制約を指定することもできる。この制約は、それらのプロパティ / 関係に対する条件上の形態のクラスに対する制約に関連し得、この制約により、構成中に使用されるマイクロ・ブループリントに関して有効な選択肢が選別される。例えば、この要求は、最低限のJ2EE適合レベルである1.5を有するJ2EEの適合スタック (compliant stack) などの制約を指定することができる。

【0060】

また、この要求は環境を指定することができる。この指定された環境が、コンポーネント又はソフトウェア・スタックの要素の選択に影響する。例えば、DEVの環境では、Tomcatのアプリケーションサーバが選択され得、一方、PRODの環境では、クラスター化されたWebLogicのアプリケーションサーバが選択され得る。

【0061】

この要求の中に含まれている情報に基づいて、マイクロ・ブループリント・アセンブラ110は、再使用可能なマイクロ・ブループリントから、1つ以上の完全なアプリケーションのブループリントを生成するよう構成される。例えば、このマイクロ・ブループリント・アセンブラ110は、要求に従って、マイクロ・ブループリントのデータベース130から、各サービス層に対して、及び各サービス層内の各スタックエレメントに対して、複数のマイクロ・ブループリントのサブセットをアセンブルすることができる。まず、このアプリケーション・ブループリント・アセンブラ115は、アプリケーションのサービス層に従って、マイクロ・ブループリントをアセンブルすることができる。ある実施形態では、このアプリケーション・ブループリント・アセンブラ115は、要求、及びマイクロ・ブループリントのデータベース130内の複数のマイクロ・ブループリントの必要な能力及び有効な能力に従って、サービス層に対応するマイクロ・ブループリントを取得することができる。サービスレベルのマイクロ・ブループリントの構成の詳細に関しては、図7を参照して、さらに説明する。さらに、マイクロ・ブループリントのデータベース130に格納される複数のマイクロ・ブループリントが比較的大きい場合、このアプリケーション・ブループリント・アセンブラ115は、AIの探索アルゴリズムを用いることができるが、これに関しては、下記にさらに詳細に説明する。

【0062】

その後、この機能コンポーネントのブループリント・アセンブラ120は、各サービス層に対してスタックエレメントに対応するマイクロ・ブループリントをアセンブルすることができる。例えば、機能コンポーネントのブループリント・アセンブラ120は、要求に従って、各サービス層に対して、スタックエレメントに対応するマイクロ・ブループリントを取得することができる。スタックエレメントに対応するマイクロ・ブループリントのアセンブリに関しては、図8を参照して、さらに詳しく説明する。

【0063】

図7は、実施形態に従った、サービス層用のマイクロ・ブループリントをアセンブルするための工程を示すフローチャートである。図7は、連続して順序付けられた動作の一覧として示されているが、これらの動作のうちのいくつか、又は全ては、異なる順序で、又は平行して、又は繰り返して行うことができる、或いは、時間を重ねて行うことができることは理解されよう。

【0064】

要求から少なくとも1つの特徴、制約、及び / 又は、環境が抽出される (702)。例えば、アプリケーション・ブループリント・アセンブラ110は、少なくとも特徴を抽出することができる。この特徴により、特定の種類のアプリケーションを指定することが可能である。しかし、この特徴には、アプリケーションのアセンブリに関連する全ての種類の特徴が含まれる可能性がある。さらに、この要求は、1つ以上の制約、及び / 又は、環境を指定

10

20

30

40

50

することができる。

【 0 0 6 5 】

この要求からリンクされたモデルクラスが抽出される(704)。例えば、このアプリケーション・ブループリント・アセンブラ110は、この要求から、リンクされたモデルクラスを抽出することができる。すなわち、この特徴は、複数のクラスのうちの1つ以上に対応することができる、制約、及び/又は、環境は、クラスプロパティのうちの1つ以上に対応することができる。つまり、このアプリケーション・ブループリント・アセンブラ100は、特徴、制約、及び環境に関する関連クラスを取得することができる。

【 0 0 6 6 】

アプリケーション・ブループリントのアセンブリ処理が呼び出される(706)。例えば、このアプリケーション・ブループリント・アセンブラ110は、アプリケーションのブループリントをアセンブルする処理を開始することができ、アプリケーション・ブループリント・アセンブラ115は、アプリケーションのサービス層に対応するマイクロ・ブループリントのアセンブリを開始することができる。

10

【 0 0 6 7 】

抽出されたクラスを有するマイクロ・ブループリントが取得される(708)。例えば、このアプリケーション・ブループリント・アセンブラ115は、抽出されたクラスを有するマイクロ・ブループリントを取得するために、マイクロ・ブループリントのデータベース130を探索することができる。例えば、アプリケーション・ブループリント・アセンブラ115は、マイクロ・ブループリントのサブセットを取得するために、マイクロ・ブループリントの必要な能力、及び有効な能力に基づいて、有効な能力及び必要な能力のうちの1つ以上が一致するように、マイクロ・ブループリントのデータベース130を探索することができる。

20

【 0 0 6 8 】

現状のマイクロ・ブループリントの注釈付のクラスを確認して、要求において指定された制約と一致しているかどうかを判定する(710)。例えば、要求に1つ以上の制約が含まれる場合、このアプリケーション・ブループリント・アセンブラ115は、要求において指定された制約と、現状のマイクロ・ブループリントに対応するモデルデータ300のクラスプロパティとが一致しているかどうかを判定する。

【 0 0 6 9 】

30

マイクロ・ブループリントを確認して、そのブループリントが選択された環境に受け入れられるかどうかを判定する(712)。例えば、アプリケーション・ブループリント・アセンブラ115は、現状のマイクロ・ブループリントが要求において指定された環境に受け入れられるかどうかを判定することができる。

【 0 0 7 0 】

マイクロ・ブループリントが取得される(714)。例えば、710と712の条件が満たされた場合、このアプリケーション・ブループリント・アセンブラ115は、マイクロ・ブループリントのデータベース130からマイクロ・ブループリントを取得することができる。この処理を、708で取得した各マイクロ・ブループリントに対して繰り返すことができる。

【 0 0 7 1 】

40

上記で説明した図7の処理は一連の命令で構成され、これらの命令が実行されると、1つ以上のプロセッサが一連の機能を実行する。これらの命令には、下記のスクリプトが含まれ得る。:

【 数 1 】

```

// Process to generate complete composed blueprint for application
for all Functional & Non Functional Features in Request {
    for all Model Classes in Features {
        Call function process(Classes, Constraints, Environment)
    }
    // Above yields a list of candidate composed complete blueprints with micro
    //blueprints for each of the related tiers or functional components
    List<Blueprint> composedBlueprints
    for all composedBlueprints {
        for each tier microBlueprint in a composedBlueprint {
            Invoke Process to generate the composed blueprint for the software
            stack of the component
            Add this to the tier microBlueprint
        }
    }
}

List<MicroBlueprint> function process(Classes, Constraints, Environment) {
    List<MicroBlueprint> microBlueprints
    for all Classes {
        microBlueprints = Find MicroBlueprint(s) in Repository annotated
        with these classes i.e.
        class is subset FunctionalComponent.Capabilities.FunctionalFeature.type
    }

    for all microBlueprints {
        Check if current MicroBlueprint annotated classes match the specified
        tier level constraints (if any)
        if(no) Remove from microBlueprints
    }
    for all microBlueprints satisfying constraints {
        Check if current MicroBlueprint is admissible in the selected
        Environment
    }
    if(no) Remove from microBlueprints

    for all microBlueprints dynamically discover the linked functional
    component micro-blueprints {
        // recursive step
        List<MicroBlueprint> linkedMicroBlueprints =
        Call function process (Current.MicroBlueprint.Required Capabilities.Classes,
        Constraints, Environment)
        if(linkedMicroBlueprints empty) Remove current from microBlueprints
        else Link linkedMicroBlueprints to current MicroBlueprint
    }
    return microBlueprints
}

```

【 0 0 7 2 】

次に、機能コンポーネントのブループリント・アセンブラ¹²⁰が、ソフトウェア・スタックの要素に対応するマイクロ・ブループリントを、各サービス層に対して、アセンブルするよう構成される。例えば、この機能コンポーネントのブループリント・アセンブラ¹²⁰は、各サービス層に対して、パッケージ/ソフトウェア・スタックの要素

を動的に構成してソフトウェア・スタックをビルドするよう構成される。

【0073】

ある実施形態では、機能コンポーネントのブループリント・アセンブラ120は、個々のソフトウェア・エレメントのマイクロ・ブループリントから好適なソフトウェア・スタックを自動でビルドする処理において、これらの必要関係、及びその限定又は制約を利用する。この必要関係は、ソフトウェア・スタックから見た、抽象的なソフトウェア・エレメントに関するクラス間の従属関係を表すものであるため、このようなことが可能である。

【0074】

図8は、実施形態に従った、ソフトウェア・スタックのエレメントに対応するマイクロ・ブループリントをアセンブルするための処理を示すフローチャートである。図8は連続して順序付けられた動作の一覧として示されているが、これらの動作のうちのいくつか、又は全ては、異なる順序で、又は平行して、又は繰り返して行うことができる、或いは、時間を重ねて行うことができることは理解されよう。

10

【0075】

ソフトウェア・スタックのエレメント用のマイクロ・ブループリントが各サービス層に対して取得され得る(802)。例えば、この機能コンポーネントのブループリント・アセンブラ120は、図7の処理で取得されたサービス層のマイクロ・ブループリントに関するマイクロ・ブループリントを取得するよう構成することができる。

【0076】

このマイクロ・ブループリントに関する注釈付のクラスが取得される(804)。例えば、機能コンポーネントのブループリント・アセンブラ120が、この取得されたマイクロ・ブループリントから関連する注釈付のクラスを取得することができる。この処理では、現状の注釈付のクラスがオペレーティングシステムのクラスかどうかを判定することができる(806)。例えば、機能コンポーネントのブループリント・アセンブラ120が、現状の注釈付のクラスがオペレーティングシステムのクラスかどうかを判定することができる。オペレーティングシステムのクラスである場合、それは最後のソフトウェア・スタックのエレメントであることを示しているため、この処理は停止する。オペレーティングシステムでない場合、この処理は810に続く。この具体的な例では、オペレーティングシステムのソフトウェアのエレメントスタックを終了ポイントとして利用しているが、本実施形態では、全ての種類のソフトウェア・スタックのエレメントを終了ポイントとして使用することができ

20

30

【0077】

関係情報により指し示されたリンクするクラスが取得される(810)。例えば、機能コンポーネントのブループリント・アセンブラ120は、必要関係により指し示された、リンクされたクラスを取得することができる。要求に含まれる特徴、制約、及び環境がそれぞれこれらのクラスに適用される(812)。例えば、機能コンポーネントのブループリント・アセンブラ120は、要求に含まれる特徴、制約、及び環境をそれぞれこれらのクラスに適用させることができる。その後、この処理は、各ソフトウェア・スタックのエレメントに対して、各サービス層で繰り返される。

【0078】

上記に説明した図8の処理は一連の命令で構成され、これらの命令が実行されると、1つ以上のプロセッサが一連の機能を実行する。これらの命令には、下記のスクリプトが含まれ得る。:

40

【数 2】

// Process to generate composed blueprint for a tier or functional component consisting of a suitable software stack

// automatically from micro-blueprints of the individual software elements

- Get app Deployment micro blueprint for Functional Component or tier e.g. for PetStore App Tier it is the PetStore EAR micro blueprint

- Get annotated class(es) for the Deployment micro blueprint e.g. for PetStore EAR it is EAR class from the model

- Carry on in a recursive manner until you reach the OS class - stop here as you have the software stack {

- For annotated class get linked class(es) pointed to by the needs* relation e.g. for EAR above the needs relation points to J2EEServer class

- If a more specialized needs* relation is present then one chooses the linked specific class - otherwise there could be many choices

for the linked classes

e.g. Deployment can be linked to any AppServer

- Apply specified class properties level constraints to filter the classes to get the matching ones e.g. a constraint like

J2EEServer.conformanceLevel > 1.5 can filter out .NET app servers

- Look at non functional requirements to filter out options that do not match them e.g. if HA is required then instead of WebLogic Blueprint

WebLogic clustered Blueprint is chosen

- Also filter for Environment e.g. Dev, Prod may have different requirements

}

10

20

【0079】

また、上記で説明したとおり、マイクロ・ブループリントの数が比較的大きい場合、アプリケーション・ブループリント・アセンブラ110は、AIの探索アルゴリズムを用いて、マイクロ・ブループリントをアセンブルすることができる。

【0080】

AI検索は、例えば、ゲーム、ロボット動作計画などの探索空間が非常に大きな様々な課題で用いられる。AIの探索アルゴリズムにより、コンピュータを用いて、探索などの一連の課題が解決し易くなる。この課題は、人工知能(AI)の探索の課題として作られており、ソフトウェア・スタック、及び層が規定されている完璧なブループリントを完全にアセンブルすることで解決する。

30

【0081】

より具体的には、この課題は、A* (AStarのAI探索アルゴリズム) 探索の課題として作られている。簡単に説明すると、このA*アルゴリズムとは、ヒューリスティックベースの「最良優先」探索アルゴリズムであり、この最良優先探索アルゴリズムでは、下記のヒューリスティックの等式に基づいて、全ての現状の状態において、その次に最良の状態を選択することにより、初期状態からスタートし、その初期状態を目標状態(複数可)に到達させる。

40

【0082】

式(1) $f(n) = g(n) + h(n)$

【0083】

上記の式のパラメータ $g(n)$ は、選択されたパス内で発生する最も低いコストであり、この選択されたパス(例えば、今のところ選択された状態から形成されているパス)で、この探索アルゴリズムは、初期状態を、今のところの現状の状態まで到達させる。特定なパスの区画のコストは、課題特有の方法で規定される。

【0084】

50

上記の式のパラメータ $h(n)$ は、現状の状態から目標状態（複数可）までの、残りのパスにおける推定コストである。これらのパラメータ、目標状態（複数可）、機能、及びコストは全て、課題の中で必要に応じて規定される。この場合、一連の能力及び一連の必要な能力を有する機能コンポーネントにより、現状の状態を表すことができる。マイクロ・ブループリントのデータベース130内には、いくつかの機能コンポーネントが存在し得、これらの機能コンポーネントが、単独で、又は組み合わせることで必要な能力を満たす。

【0085】

例えば、具体的に例を挙げると、「ウェブ層」のコンポーネントが、現状の状態「状態 n 」を形成する。このウェブ層のコンポーネントは、必要な能力として「商品目録」及び「ショッピングカート」を有する。したがって、マイクロ・ブループリントのデータベース130内をクエリー検索することにより、3つのコンポーネントをもたらすことができ、これらの3つのコンポーネントにより、これらの必要な能力、すなわち「アプリケーション層」（完全に要求事項を満たす）、「商品目録サービス」（部分的に要求事項を満たす）、及び「ショッピングカートサービス」（部分的に要求事項を満たす）が部分的に、又は完全に満たされる。各状態により現状の状態「状態 n 」の要求事項が完全に満たされるように、これらの能力から、次の状態の候補（例えば、「状態 $n+1$ 」及び「状態 $n+2$ 」）を生成する。この場合、「状態 $n+1$ 」は「商品目録サービス」のコンポーネント及び「ショッピングカートサービス」のコンポーネントから構成され、「状態 $n+2$ 」は「アプリケーション層」のコンポーネントから構成される。

【0086】

次に、これらの状態の中から最適な状態が最適状態として選択される。例えば、制約も考慮して最小の数のコンポーネントしか伴わない、ヒューリスティック同様の選択状態がこのコストには含まれ得る。この最適状態はコスト関数 $g(n)$ にも供給される。したがって、最も低いコストのパス、すなわち最良のパスがビルドされる。

【0087】

それまでに蓄積された選択状態の「能力」及び蓄積された「必要な能力」の記録をとることにより、目標状態に到達する。蓄積された「必要な能力」が、蓄積された「能力」のサブセットになると、完全なブループリントと判定され、この処理は探索を停止する。

【0088】

結果として、本実施形態のデータ処理装置は、ブループリントを構成可能なマイクロ・ブループリントと再使用可能なマイクロ・ブループリントとに分解することができ、これらのマイクロ・ブループリントには、モデルのデータベース135内のモデルデータからの1つ以上のクラスの注釈を付けられる。基本的には、これらの注釈付のマイクロ・ブループリントにより、サービス、又はソフトウェア・スタックのマイクロ・ブループリントが1つ以上のモデルデータのクラスに割り当てられ、このモデルデータには、クラス間の関係を規定する関係情報も含まれる。注釈付のマイクロ・ブループリントとモデルデータを用いることにより、このデータ処理装置は、マイクロ・ブループリントから、その場で完全に構成されたブループリントを自動的にアセンブルし、これにより、アプリケーションの供給、すなわち開発が可能となる。すなわち、データ処理装置100は、モデル駆動型アプローチを行って、マイクロ・ブループリントなどの小さなビルディング・ブロックから、完全なブループリントを自動で柔軟且つ動的に構成する。上記に示すとおり、この完全なアプリケーションのブループリントには、機能ブループリント、又は配置ブループリントが含まれ得る。この完全なブループリントを用いて、そのコンポーネント又は層と、それらの各ソフトウェア・スタックの要素とを含む完全なアプリケーションを開発することができる。

【0089】

本明細書に記載されている種々の技術の実装形態は、デジタル電子回路内、又はコンピュータのハードウェア、ファームウェア、ソフトウェア内、或いはそれらの組合せ内で実行可能である。実装形態は、コンピュータプログラム製品、すなわちマシン可読記憶装置内、又は伝播信号内などの情報担体内で有形に具現化されるコンピュータプログラムとし

てとして実行可能であり、このコンピュータプログラム製品は、プログラマブルプロセッサ、コンピュータ、又は複数のコンピュータなどのデータ処理装置の動作により実行される、或いは、それらの動作を制御する。上記で説明したコンピュータプログラム（複数可）などのコンピュータプログラムは、コンパイル型言語又はインタープリタ型言語を含む全ての形態のプログラミング言語で書き込むことができ、独立プログラム、或いはモジュール、コンポーネント、サブルーチン、又はコンピュータ環境内での使用に適したその他のユニットを含む全ての形態で配置可能である。コンピュータプログラムは、単独のコンピュータ上、又は1つの場所に配置される、又は複数の場所に配置される、通信ネットワークで相互接続する、複数のコンピュータ上に配置可能である。

【0090】

10

方法の工程は、入力データ上で動作し出力を生成することにより機能を実行するためのコンピュータプログラムを実行する1つ以上のプログラマブルプロセッサにより実行可能である。方法の工程はまた、例えば、FPGA（フィールドプログラマブルゲートアレイ）、又はASIC（特定用途向け集積回路）などの専用論理回路、及びそれらの専用理論回路として実装可能な装置によっても実行可能である。

【0091】

コンピュータプログラムを実行するための好適なプロセッサには、例えば、汎用マイクロプロセッサ、及び専用マイクロプロセッサの両方、並びに全ての種類のデジタルコンピュータの1つ以上の全てのプロセッサが含まれる。通常、プロセッサは、リード・オンリー・メモリ又はランダム・アクセス・メモリ、或いはそれらの両方から命令及びデータを受け取る。コンピュータの構成要素には、命令を実行するための少なくとも1つのプロセッサ、及び命令とデータを格納するための1つ以上の記憶装置が含まれ得る。通常、コンピュータは、データを格納するための1つ以上の大容量記憶装置、例えば、磁気ディスク、光磁気ディスク、又は光ディスクを備えることができる。或いは、こういった1つ以上の大容量記憶装置に動作可能に接続して、データの受信又は送信、或いはその両方を行うことができる。コンピュータプログラムの命令及びデータを包含するために好適な情報担体には、全ての形態の不揮発性メモリが含まれ、これらの不揮発性メモリをいくつか例示すると、半導体記憶装置（例えば、EPROM、EEPROM、及びフラッシュメモリ装置）、磁気ディスク（例えば、内部ハードディスク、又は取り外し可能ディスク）、光磁気ディスク、並びにCD-ROMディスク及びDVD-ROMディスクなどが挙げられる。これらのプロセッサ及びメモリは、専用論理回路により補われ得る、あるいは専用論理回路に組み込まれ得る。

20

30

【0092】

ユーザとのやりとりを可能にするために、情報をユーザに表示するためのブラウン管（CRT）のモニタ、又は液晶表示装置（LCD）のモニタなどの表示装置、ユーザのコンピュータへの入力を可能にするキーボード及びマウス又はトラックボールなどのポインティングデバイスを備えるコンピュータ上で、実装形態を実行することができる。その他の種類の装置を用いて、ユーザとのやり取りを行うことも可能である。それらのやり取りの例として、視覚フィードバック、聴覚フィードバック、又は触覚フィードバックなどの全ての形態でユーザに供給可能な感覚フィードバックや、音による入力、口頭による入力、触感による入力を含む全ての形態でユーザから受け取る入力などが挙げられる。

40

【0093】

データサーバなどのバックエンドコンポーネント、アプリケーションサーバなどのミドルウェアコンポーネント、又はクライアントコンピュータなどのフロントコンポーネントを備えるコンピュータシステム内で実装形態を実行することができる。クライアントコンピュータは、グラフィカルユーザインターフェース、又はウェブブラウザを備え、これらを通してユーザは実装形態、又はこのようなバックエンドコンポーネント、ミドルウェア、又はフロントエンドコンポーネントの全ての組合せとやり取りすることが可能である。これらのコンポーネントは、デジタルデータ通信の全ての形態又は媒体、例えば、通信ネットワークにより相互接続が可能である。通信ネットワークの例としては、ローカル・エリア・ネットワーク（LAN）、又はインターネットなどのワイド・エリア・ネットワーク

50

(WAN) が挙げられる。

【 0 0 9 4 】

記載されている実装形態のいくつかの特徴を本明細書に記載されているとおり説明してきたが、多くの修正形態、代替形態、変更形態、及び等価形態が当業者により考案される。

【 図 1 】

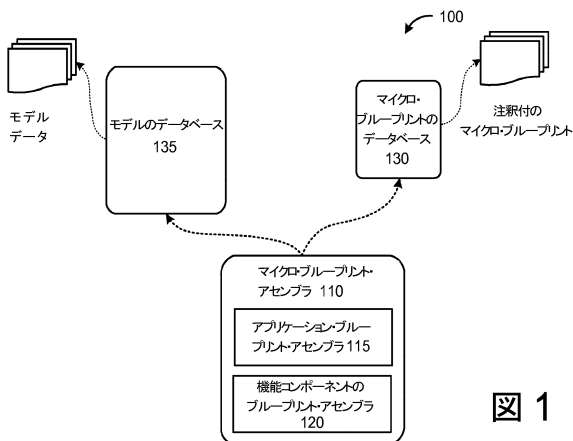


図 1

【 図 2 】

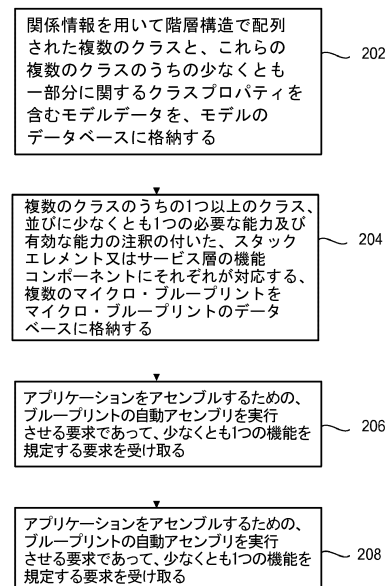


図 2

【図 3】

300

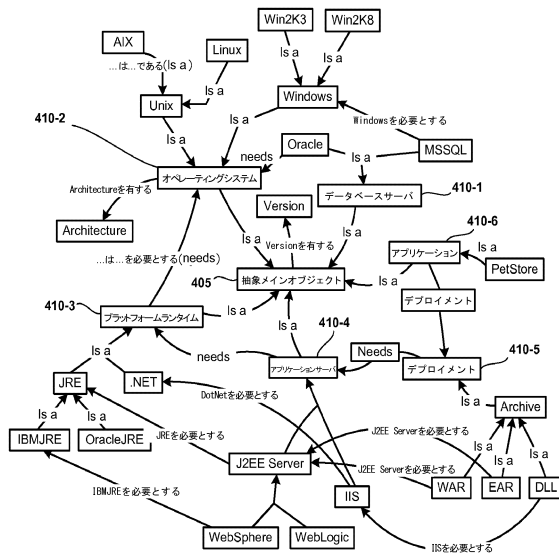


図 3

【図 4】

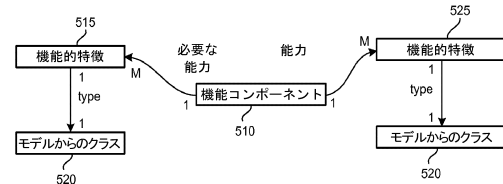


図 4

【図 5】

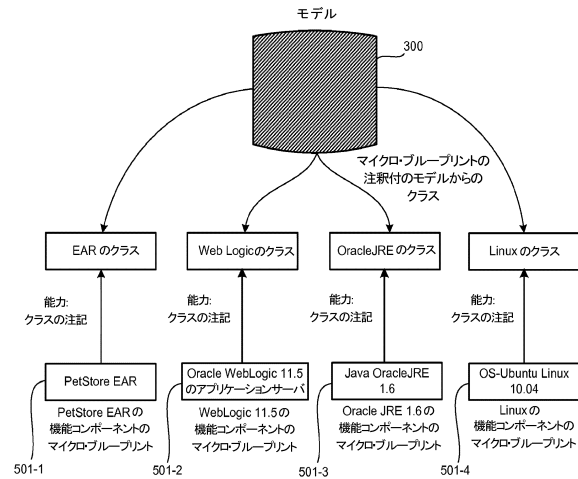


図 5

【図 6】

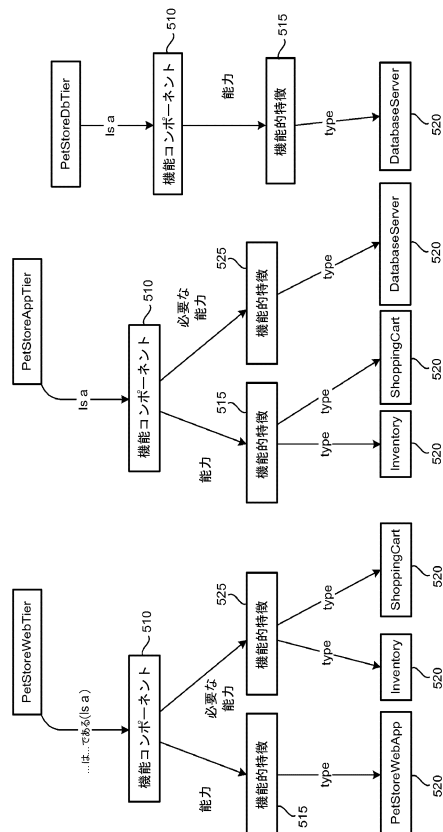


図 6

【図 7】

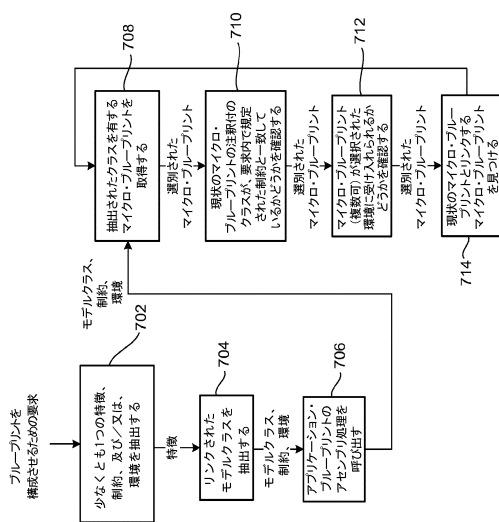


図 7

【図 8】

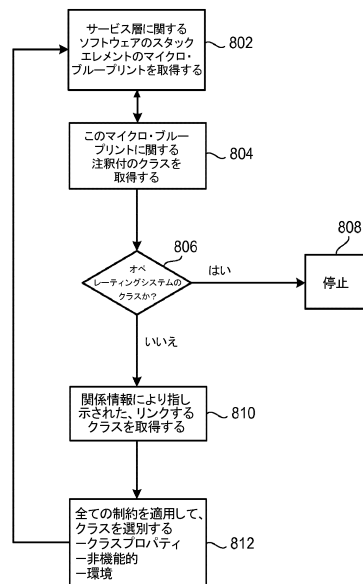


図 8

フロントページの続き

(72)発明者 ネエラン カルニク

インド国 プネ 411008 パネル ロード イボリ エスタテス ティアラ エー 502

(72)発明者 アブハイ ガシサス

インド国 プネ 411038 コスラド シバテエルス ナガル ジェイ 201 サムラジャ

審査官 杉浦 孝光

(56)参考文献 特開2009-251769(JP, A)

米国特許出願公開第2010/0031247(US, A1)

米国特許出願公開第2004/0268296(US, A1)

(58)調査した分野(Int.Cl., DB名)

G06F 8/70