(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
12 January 2017 (12.01.2017)

WIPO | PCT

(10) International Publication Number
WO 2017/007466 A1

(51) International Patent Classification:
G06Q 10/06 (2012.01)

(21) International Application Number:
PCT/US2015/039526

(22) International Filing Date:
8 July 2015 (08.07.2015)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **HEWLETT PACKARD ENTERPRISE DEVELOPMENT LP** [US/US]; 11445 Compaq Center Drive West, Houston, TX 77070 (US).

(72) Inventors: **MARNEY, Steven H**; 585 South Boulevard, Pontiac, Michigan 48341 (US). **KAMALAKANTHA, Chandra H.**; 5400 Legacy, Plano, Texas 75024 (US). **DOSHI, Parag**; 5555 Windward Pkwy, Alpharetta, Georgia 30004 (US).

(74) Agents: **FEBBO, Michael A.** et al.; Hewlett Packard Enterprise, 3404 E. Harmony Road, Mail Stop 79, Fort Collins, CO 80528 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to the identity of the inventor (Rule 4.17(i))*

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

Published:

— *with international search report (Art. 21(3))*

(54) Title: ORCHESTRATION TEMPLATE GENERATION

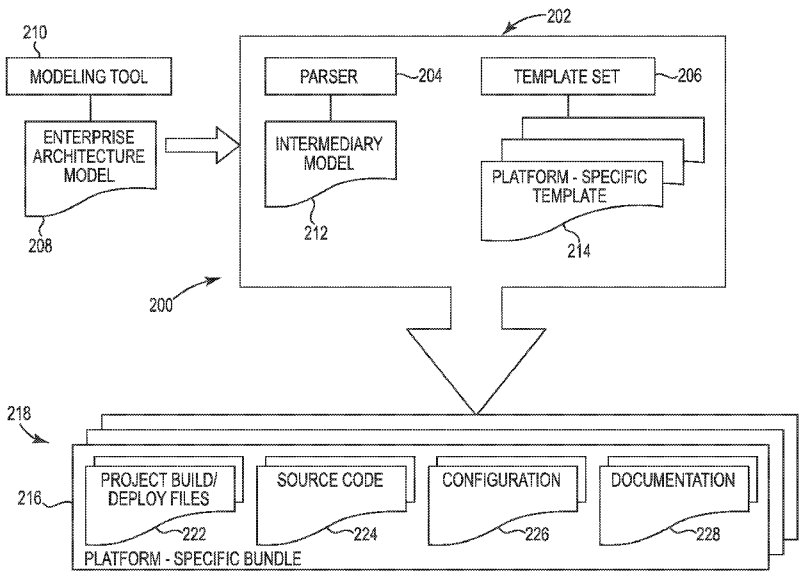

Fig. 2

(57) Abstract: Orchestration template generation for a computer network such as cloud environment is described. A platform-independent intermediary model is generated from an enterprise architecture model. The intermediary model is applied to a selected platform-specific deployment configuration template associated with a selected deployment format.

WO 2017/007466 A1

ORCHESTRATION TEMPLATE GENERATION

Background

[0001] Cloud computing is a model of service delivery for enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with nominal management effort or interaction with a provider of the service. Cloud computing allows a consumer to obtain computing resources, such as networks, network bandwidth, servers, processing memory, storage, applications, virtual machines, and services as a service on an elastic and sometimes impermanent basis. The creation, management, manipulation, and decommissioning of computing resources in order to realize consumer computing requests while conforming to the operational objectives of cloud service providers is known as resource orchestration. Resource management tools use orchestration templates to automate deployments for resource orchestration.

Brief Description of the Drawings

[0002] Figure 1 is a schematic diagram illustrating an example cloud computing environment.

[0003] Figure 2 is a schematic diagram illustrating an example system for generating orchestration templates in the example cloud environment of Figure 1.

[0004] Figure 3 is a schematic diagram illustrating an example domain model constructed as a class diagram for use in the system of Figure 2.

[0005] Figure 4 is a block diagram illustrating a method of the system of Figure 2.

[0006] Figure 5 is a schematic diagram illustrating an example computing device that can be used to implement the system of Figure 2 and perform the method of Figure 4.

Detailed Description

[0007] In the following detailed description, reference is made to the accompanying drawings which form a part hereof, and in which is shown by way of illustration specific examples in which the disclosure may be practiced. It is to be understood that other examples may be utilized and structural or logical changes may be made without departing from the scope of the present disclosure. The following detailed description, therefore, is not to be taken in a limiting sense, and the scope of the present disclosure is defined by the appended claims. It is to be understood that features of the various examples described herein may be combined, in part or whole, with each other, unless specifically noted otherwise.

[0008] Cloud providers can use cloud service management, or enterprise management, tools to automate the management of cloud-based IT-as-a-service from order, to provision, to retirement. Traditionally, IT (information technology) professionals manually patched and updated physical servers. Cloud service management tools accelerate the speed of deployment of application-based services across many different forms of cloud environments. Also, cloud service management tools can be used to move and manage applications among cloud systems, provide security and consistency, and dynamically monitor and provision server, storage, and network elements. IT professionals often use cloud service management tools that allow them to model the deployment topology of distributed systems and heterogeneous computing environments and use orchestration workflows to automate their deployment and configuration. An example of a cloud service management tool is available under the trade designation Cloud Service Automation from Hewlett-Packard of Palo Alto, California.

[0009] Cloud service management tools can also be used to automate the creation of cloud environments, which can begin as soon as the racks and servers of a data center are operational. Cloud service management tools often enable automation logic to be orchestrated via workflows that sequence the

automation steps to be performed. Orchestration templates are used in the automation process to describe infrastructure and applications for a cloud service management tool in a manner that is often also readable and writable by humans.

[0010] Prior to modeling deployment topologies in the cloud services management tool, many IT professionals also use modeling tools to create a visualization in terms of an enterprise architecture model. Enterprise architecture models include domain models that comprise business architectures and system models of the cloud environment and services running in the cloud environment as complex systems. A typical process of creating a new system available from an ordering catalog can take months of painstaking crafting skills to generate an enterprise architecture model. IT professionals, using another set of painstaking crafting skills, then covert enterprise architecture specifications of the complex systems into a set of orchestration specifications that the cloud service management tools use to drive lifecycle automation. Recreating the enterprise architecture model and its associated orchestration workflows in a cloud service management tool is time consuming and error prone and made more difficult when the domain is fluid.

[0011] An example heuristic system and method to interpret and automatically convert the enterprise architecture model and generate orchestration templates to enable automated deployment in a computer network such as a cloud environment is described. An example system includes an engine having a parser and a set of code generation templates. An enterprise architecture model, which may be created with a modeling tool, is received at the parser, which generates a platform-independent intermediary model. The intermediary model is applied to a selected platform-specific deployment configuration template to generate an orchestration template, which may include a deployment bundle or orchestration logic, in a selected deployment format that can be applied to resource management tool and used to automate deployment of the enterprise architecture. In one example, the system includes several available platform-specific deployment configuration templates, and user can choose one or more of the templates to generate one or more deployment

bundles in one or more deployment formats. The system and method allow for enterprise architecture models in traditional or common formats, for example, to automate deployment in one or more orchestration formats.

[0012] Figure 1 illustrates an example cloud computing environment 100 suitable for use with the system and method to generate orchestration templates. Cloud computing environment 100 includes one or more interconnected cloud computing nodes 102 configured to communicate with local computing devices 104 such as personal computers, mobile devices, embedded systems, or other computing devices used by cloud consumers. Cloud computing environment 100 includes features such as statelessness, low coupling, modularity, and semantic interoperability. Cloud computing nodes 102 can be configured as computing devices including a processor, memory, storage, communication components, and software in the form of program modules stored in the memory. Cloud computing nodes 102 may be grouped physically or virtually in one or more networks or in one or more cloud deployment models. The cloud computing environment 100 offers services such as infrastructure, platforms, software, and business processes.

[0013] Cloud computing environment 100 can include a set of abstraction layers such as a hardware and software layer 106, virtualization layer 108, management layer 110, and workload layer 112. The hardware and software layer 106 includes hardware and software components such as servers, storage devices, networking and networking components, network application software, database software, and related software. The virtualization layer 108 provides virtualization entities such as virtual servers, storage, networks, and applications. The management layer 110 provides entities such as resource provisioning, metering and billing services for tracking and invoicing use, user portals for allowing cloud consumers and others access to the cloud computing environment 100, security, and service level management. Workload layer 112 provides functions such as mapping and navigation, software development and lifecycle management, data processing, and transaction processing. The components, layers, and other features of the cloud computing environment 100

are intended to be illustrative, and other example configurations are contemplated.

[0014] Figure 2 illustrates an example system for generating orchestration templates 200 for use in automating the cloud environment 100 or other computer network. The system 200 includes an engine 202 having a parser 204 and a template set 206. The engine 202 can incorporate features of a template engine, which is designed to combine one or more templates with a data model to produce one or more result documents. The system 200 is configured to receive an enterprise architecture model 208 produced with a modeling tool 210, such as a system described as a visualization in terms of business architecture. The parser 204 creates an intermediary model 212 from the enterprise architecture model 208. The engine 202 applies the intermediary model 212 to one or more selected platform-specific templates 214. The system 200 generates a platform-specific orchestration template, which can include deployment bundle 216, in one or more selected deployment formats 218. The platform-specific deployment bundle 216 includes can then be used with a cloud services management tool to automate deployment in a process automation solution on a computer network, data center, or cloud environment.

[0015] In one example, the system 200 can be a stand-alone tool used in connection with the modeling tool 210 and the cloud services management tool configured to apply the orchestration template. In another example, the system 200 can be combined with, or incorporated into, the modeling tool 210, the cloud services management tool, or both, to become an integrated enterprise architecture to automation tool or environment.

[0016] The enterprise architecture model 208 describes a cloud environment or computing system in terms of enterprise architecture specifications. For example, the enterprise architecture model 208 can include a system model as a visualization in terms of business architecture including components and component relationships. The model of components and component relationships can be constructed in any one or more of many modeling languages to express the computing system in a structure according to a set of rules. For example, the modeling language may be described in universal

modeling language (UML) or in another modeling language such as ArchiMate. Examples of commercially-available tools that can be used as modeling tool 210 include an enterprise architecture tool such as one available under the trade designation Enterprise Architect from Sparx Systems, a diagramming and vector graphics application such as one available under the trade designation Microsoft Visio from Microsoft Corporation, or other tool for generating graphical or visual system models.

[0017] The enterprise architecture model 208 is imported into the parser 204 in one of several suitable formats compatible with the engine 202. Examples of compatible formats can include XML (Extensible Markup Language) format, such as XML Metadata Interchange format and ArchiMate format. XML Metadata Interchange, or XMI, is a common interchange format used with UML. The example of Enterprise Architect as a modeling tool 210 is capable of exporting the enterprise architecture model 208 in either XMI or ArchiMate formats. System 200 can be constructed to include a component or module to convert the enterprise architecture model 208 into a suitable format for the parser 204 in cases where the model is not provided in a compatible format.

[0018] The parser 204 receives the enterprise architecture model 208 in the engine-compatible format and generates an intermediary model 212 in a platform-independent layout or format. For example, the intermediary model layout could be limited to information regarding classes and associations to generate components and corresponding operations. For example, the parser 204 would add metadata and calculated values to the model and remove extraneous information related to comments, the particular appearance, position, or arrangement of the components, and other user-interface attributes from the enterprise architecture model 208. In one example, the parser 204 can be based on a general-purpose text-file preprocessing tool such as FMPP, which is a FreeMarker-based file preprocessor. The platform-independent intermediary model 212 is generally not configured for use directly as an orchestration template in a cloud service management tool.

[0019] Template set 206 can include one, but typically more than one, component templates 214. Each component template 214 can correspond to a

deployment format of an associated cloud management service tool designated to receive the orchestration template. The platform-independent intermediary model is applied to a selected component template based on the desired deployment bundle platform. For example, the template set 206 could include a platform-specific template 214 for a deployment bundle as a content pack for a deployment format for an automation tool available under the trade designation Operation Orchestration from Hewlett-Packard. Other supported deployment bundles can include Chef cookbooks, which are units of configuration and policy distribution for configuration management tool available from Chef, Ansible playbooks, with configuration, deployment, and orchestration bundles available from Ansible, Inc., or other cloud or enterprise configuration management tools or tools used for automation of cloud or enterprise environments.

[0020] In one example, the templates are in a format such as FreeMarker Template Language (FTL) configured to run on FreeMarker, which is a Java-based template engine. The platform-specific templates 214 can be reverse engineered configuration templates. The template engine incorporated into engine 202 applies the intermediary model 212 to the selected template to generate a deployment bundle 216 including deployment scripts or automated deployment files in compliance with an orchestration language. An example orchestration language can include TOSCA, or Topology and Orchestration Specification for Cloud Applications, an XML-based modeling language formalizing application structure as a typed topology graph. Other templates can include YAML (for example, used with Ansible), Amazon Cloud Formation templates. In one example, the deployment bundle 216 can include project build/deploy files 222, source code files 224 such as Java code (.java) or business process management notation files (.bpmn), configuration parameters 226 (such as .xml, .conf, and .properties files), and documentation 228 in text or html files.

[0021] Figure 3 illustrates an example enterprise architecture model in the form of a system model diagram 300 generated with the modeling tool 210. Figure 3 is presented to illustrate the concepts of a system model diagram 300, which are often more complex. A representation of the system model diagram 300,

such as an XML file or other set of enterprise architecture specifications, can be received into the system 200 of Figure 2 as enterprise architecture model 208. Examples of systems models can include entity-relationship models, domain models, use-case models, component models, class diagrams, and others. System model diagram 300 includes an application 302 associated with server 304 in the form of a structural UML diagram. For example, system model diagram 300 is a class diagram that captures basic information about objects 306 in the domain 308. For example, application 302 includes name of the class 310, attributes of the class 312, and methods that can operate on the class 314. Server 304 includes the class name 316 and attributes 318. The domain model can include stereotypes, which is an extensibility mechanism in UML, to target objects or operations. For example, the name 310 can include a stereotype to target the object as a component in a particular cloud service management tool, << CSA Component >>. The method 314 can include a stereotype to target operations in a particular automation tool to generate stubs for scripting secure shell, << SSH >> and ReSTful calls, << ReSTful>>.

[0022] Figure 4 illustrates an example method 400 for generating orchestration templates with a system such as the system 200 of Figure 2. Method 400 includes receiving an enterprise architecture model, such as a system model in a visualization format, at 402. In an example where method 400 is implemented with a processor, receiving can be performed with an input module. Method 400 includes generating a platform-independent intermediary model from the enterprise architecture model at 404. In one example, generating the platform-independent intermediary model includes parsing the enterprise architecture model to generate the platform-independent intermediary model, such as with a preprocessor or parser 204 of system 200. Method 400 further includes applying the intermediary model to a selected deployment configuration template at 406. The selected deployment configuration template is associated with a deployment format. Method 400 can be used to generate a deployment bundle in a selected deployment format or be used to generate multiple orchestration templates or deployment bundles in multiple selected deployment formats.

[0023] Figure 5 illustrates an example computer system that can be employed in an operating environment and used to host or run a computer application. The computer application can implement example method 400 as included on one or more computer readable storage mediums storing computer executable instructions for controlling the computer system, such as a computing device. In one example, the computer system of Figure 5 can be used to implement the modules and its associated tools set forth in system 200.

[0024] The exemplary computer system of Figure 5 includes a computing device, such as computing device 500. Computing device 500 typically includes one or more processors 502 and memory 504. The processors 502 may include two or more processing cores on a chip or two or more processor chips. In some examples, the computing device 500 can also have one or more additional processing or specialized processors (not shown), such as a graphics processor for general-purpose computing on graphics processor units, to perform processing functions offloaded from the processor 502. Memory 504 may be arranged in a hierarchy and may include one or more levels of cache. Memory 504 may be volatile (such as random access memory (RAM)), non-volatile (such as read only memory (ROM), flash memory, etc.), or some combination of the two. The computing device 500 can take one or more of several forms. Such forms include a tablet, a personal computer, a workstation, a server, a handheld device, a consumer electronic device (such as a video game console or a digital video recorder), or other, and can be a stand-alone device or configured as part of a computer network, computer cluster, cloud services infrastructure, or other.

[0025] Computing device 500 may also include additional storage 508. Storage 508 may be removable and/or non-removable and can include magnetic or optical disks or solid-state memory, or flash storage devices. Computer storage media includes volatile and nonvolatile, removable and non-removable media implemented in any suitable method or technology for storage of information such as computer readable instructions, data structures, program modules or other data. A propagating signal by itself does not qualify as storage media.

[0026] Computing device 500 often includes one or more input and/or output connections, such as USB connections, display ports, proprietary connections, and others to connect to various devices to receive and/or provide inputs and outputs. Input devices 510 may include devices such as keyboard, pointing device (e.g., mouse), pen, voice input device, touch input device, or other. Output devices 512 may include devices such as a display, speakers, printer, or the like. Computing device 500 often includes one or more communication connections 514 that allow computing device 500 to communicate with other computers/applications 516. Example communication connections can include, but are not limited to, an Ethernet interface, a wireless interface, a bus interface, a storage area network interface, a proprietary interface. The communication connections can be used to couple the computing device 500 to a computer network 518, which is a collection of computing devices and possibly other devices interconnected by communications channels that facilitate communications and allows sharing of resources and information among interconnected devices. Examples of computer networks include a local area network, a wide area network, the Internet, or other network.

[0027] Computing device 500 can be configured to run an operating system software program and one or more computer applications, which make up a system platform. A computer application configured to execute on the computing device 500 is typically provided as set of instructions written in a programming language. A computer application configured to execute on the computing device 500 includes at least one computing process (or computing task), which is an executing program. Each computing process provides the computing resources to execute the program.

[0028] Although specific examples have been illustrated and described herein, a variety of alternate and/or equivalent implementations may be substituted for the specific examples shown and described without departing from the scope of the present disclosure. This application is intended to cover any adaptations or variations of the specific examples discussed herein. Therefore, it is intended that this disclosure be limited only by the claims and the equivalents thereof.

CLAIMS

1. A method to generate orchestration templates, the method comprising:

generating a platform-independent intermediary model from an enterprise architecture model; and

applying the intermediary model to a selected deployment configuration template associated with a selected deployment format.

2. The method of claim 1 wherein the enterprise architecture model is a system model in a visual format.

3. The method of claim 2 wherein the visualization format includes a universal modeling language.

4. The method of claim 3 wherein the system model is a domain model class diagram.

5. The method of claim 1 wherein generating the platform-independent intermediary model includes parsing the enterprise architecture model.

6. The method of claim 1 wherein generating the platform-independent model includes adding metadata and calculated values to the enterprise architecture model, and removing information related to comments, appearance, and user-interface attributes from the enterprise architecture model.

7. The method of claim 1 wherein orchestration template includes a deployment bundle having automated deployment files.

8. The method of claim 7 wherein the automated deployment files are in an orchestration language.

9. A computer readable medium for storing computer executable instructions for controlling a computing device to perform a method to generate orchestration templates, the method comprising:

receiving a enterprise architecture model;

parsing the enterprise architecture model to generate a platform-independent intermediary model; and

applying the intermediary model to a plurality of selected deployment configuration templates associated with a plurality of selected deployment formats.

10. The computer readable medium of claim 9 wherein the deployment bundles include project build/deploy files, source code files, configuration parameters, and documentation files.

11. The computer readable medium of claim 9 wherein the enterprise architecture model includes a system model having stereotypes to target objects as a component in a cloud service management tool.

12. A system to generate orchestration templates from a enterprise architecture model, the system comprising:

a parser to generate a platform-independent intermediary model from the enterprise architecture model;

a selected deployment configuration template associated with a deployment format; and

an engine to apply the selected deployment configuration template to generate the orchestration template in the deployment format.

13. The system of claim 12 wherein the parser includes a text-file preprocessing tool.

14. The system of claim 12 wherein the selected deployment configuration template is a reverse engineered configuration template.

15. The system of claim 12 wherein the engine includes a template engine.

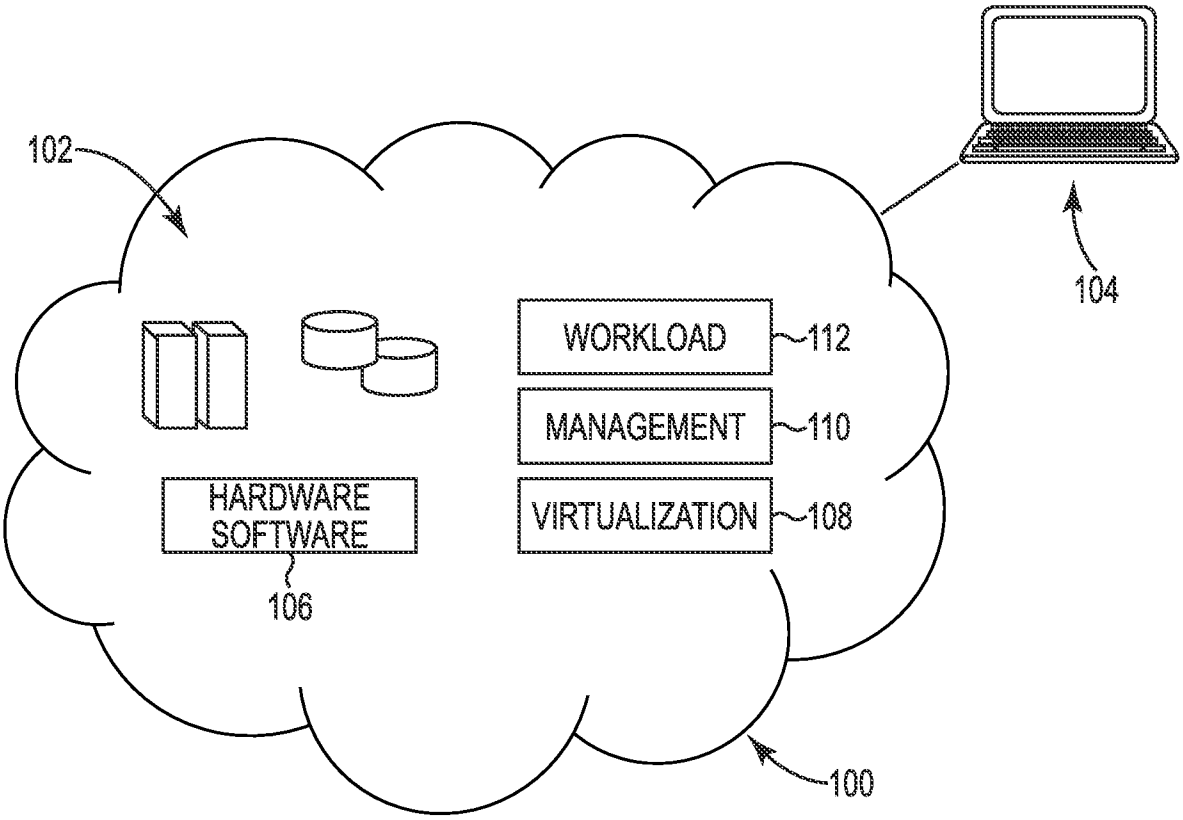
102
104
WORKLOAD
112
MANAGEMENT
110
HARDWARE
SOFTWARE
106
VIRTUALIZATION
108
100

Fig. 1

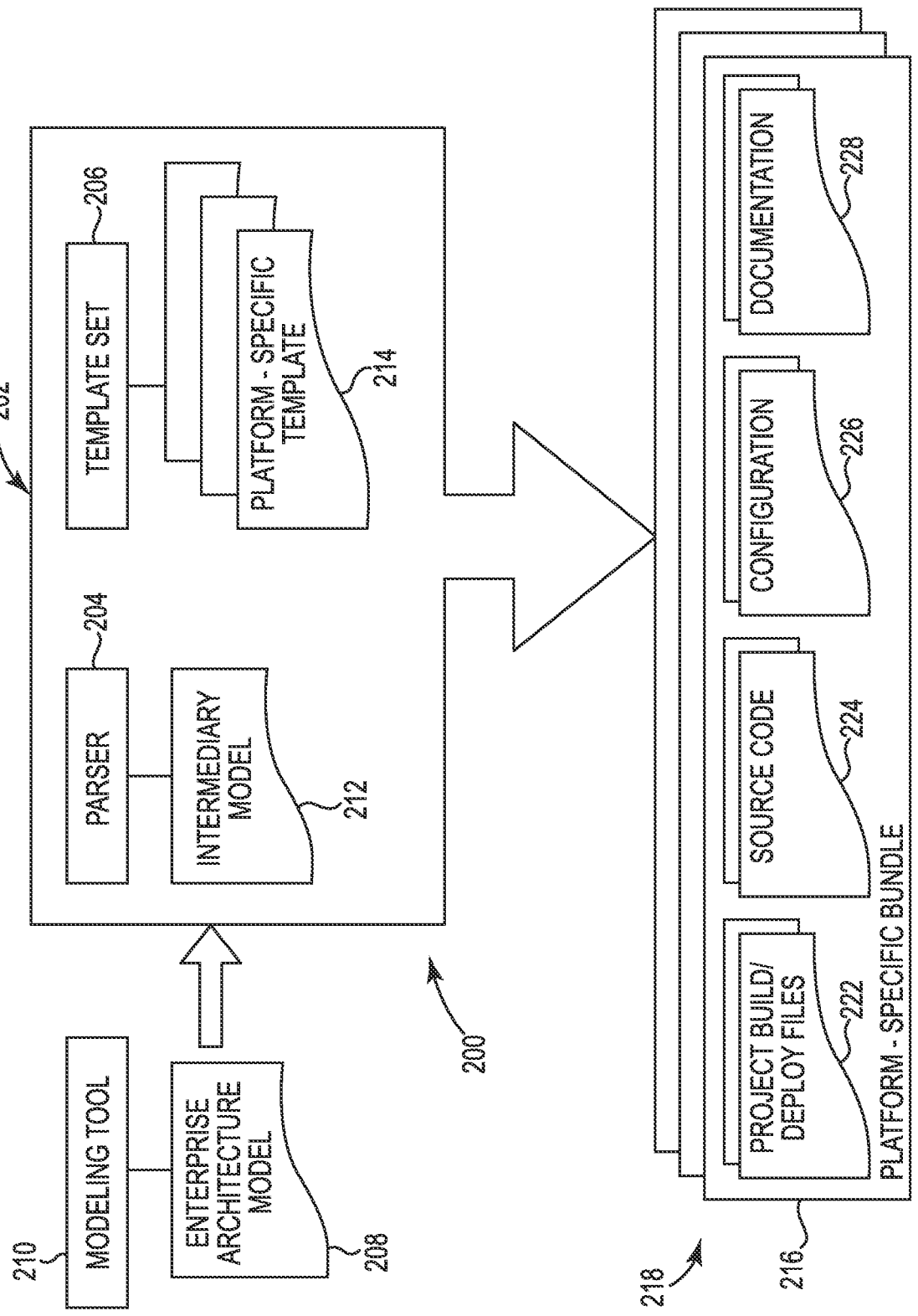
202
TEMPLATE SET
206
PLATFORM - SPECIFIC TEMPLATE
214
PARSER
204
INTERMEDIARY MODEL
212
MODELING TOOL
210
ENTERPRISE ARCHITECTURE MODEL
208
200
PROJECT BUILD/ DEPLOY FILES
222
SOURCE CODE
224
CONFIGURATION
226
DOCUMENTATION
228
PLATFORM - SPECIFIC BUNDLE
216
218

Fig. 2

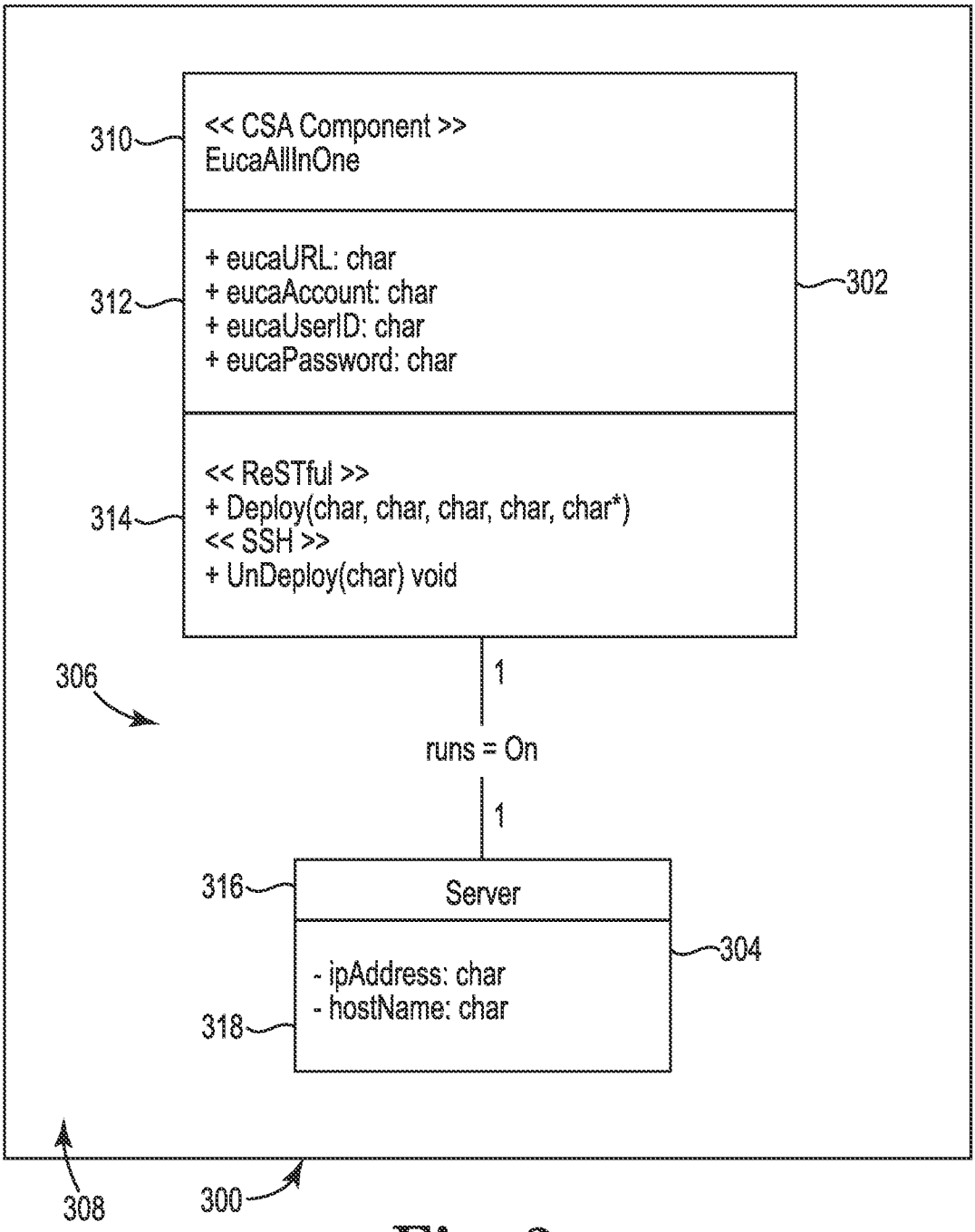
310
<< CSA Component >>
EucaAllInOne
312
+ eucaURL: char
+ eucaAccount: char
+ eucaUserID: char
+ eucaPassword: char
302
314
<< ReSTful >>
+ Deploy(char, char, char, char, char*)
<< SSH >>
+ UnDeploy(char) void
306
1
runs = On
1
316
Server
304
- ipAddress: char
- hostName: char
318
308
300

Fig. 3

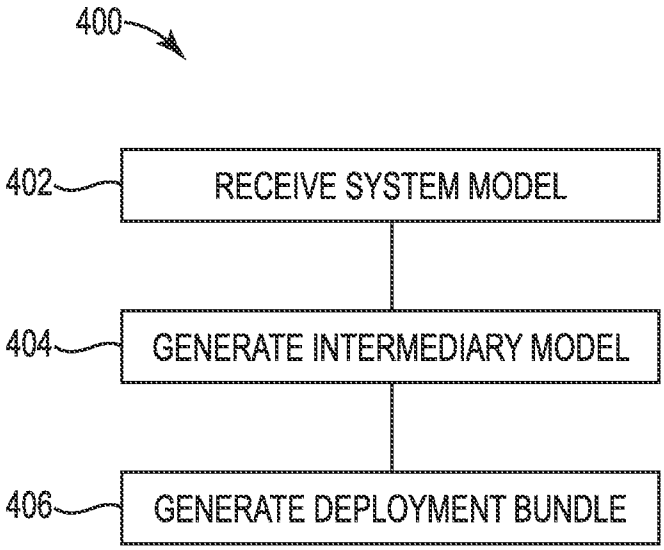
400
402
RECEIVE SYSTEM MODEL
404
GENERATE INTERMEDIARY MODEL
406
GENERATE DEPLOYMENT BUNDLE

Fig. 4

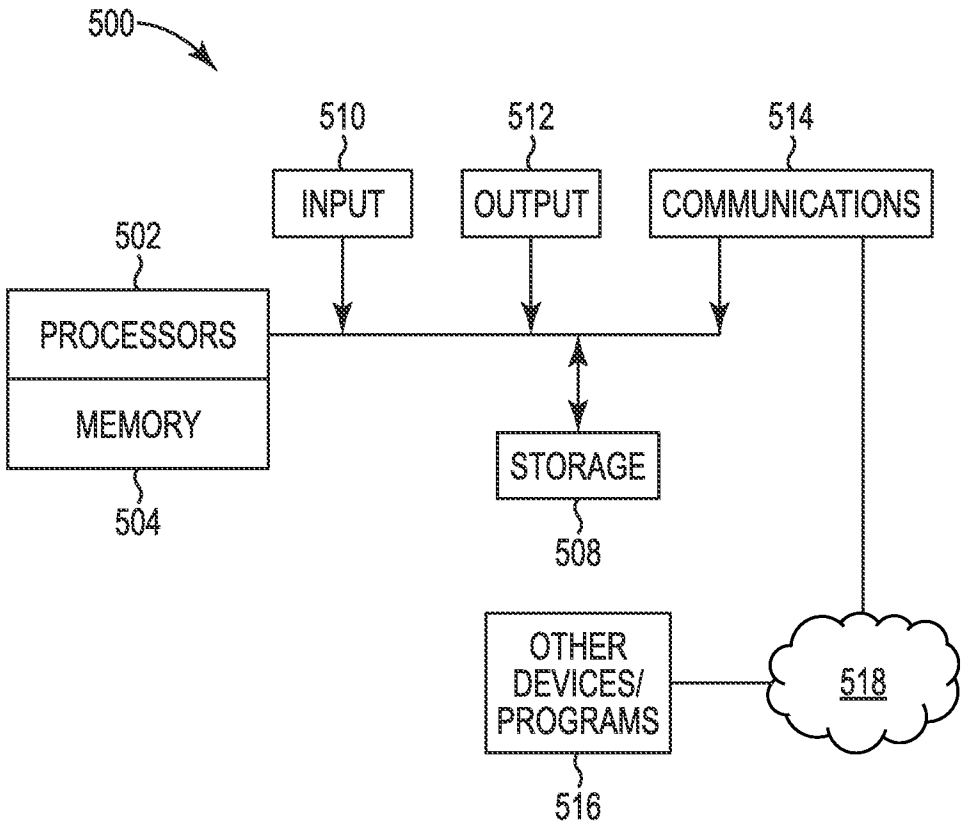
500
510
512
514
INPUT
OUTPUT
COMMUNICATIONS
502
PROCESSORS
MEMORY
504
STORAGE
508
OTHER DEVICES/ PROGRAMS
516
518

Fig. 5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2015/039526

A. CLASSIFICATION OF SUBJECT MATTER

G06Q 10/06(2012.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06Q 10/06; G06N 3/12; G06F 9/445; G06F 15/173; G06F 17/30

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean utility models and applications for utility models
Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords: enterprise architecture model, platform-independent intermediary model, deployment configuration template

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2013-0212129 A1 (ROCKWELL AUTOMATION TECHNOLOGIES, INC.) 15 August 2013 See abstract, paragraphs [0044]-[0047],[0054],[0063] and claims 1,12,13.	1-15
Y	US 2015-0006878 A1 (MICROSOFT CORPORATION) 01 January 2015 See abstract, paragraph [0034] and claims 1,2,12-14.	1-15
Y	WO 2015-094196 A1 (HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P.) 25 June 2015 See abstract and claim 1.	1-15
A	US 2012-0124211 A1 (SEAN ROBERT KAMPAS et al.) 17 May 2012 See abstract and claims 1,3,13.	1-15
A	US 2012-0072597 A1 (GREGORY WRAY TEATHER et al.) 22 March 2012 See abstract and claims 1-6,9.	1-15

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents,such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
30 March 2016 (30.03.2016)	**31 March 2016 (31.03.2016)**

Name and mailing address of the ISA/KR
International Application Division
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon, 35208, Republic of Korea
Facsimile No. +82-42-481-8578

Authorized officer
KANG, Min Jeong
Telephone No. +82-42-481-8131

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2015/039526

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2013-0212129 A1	15/08/2013	US 2013-0211546 A1	15/08/2013
		US 2013-0211555 A1	15/08/2013
		US 2013-0212160 A1	15/08/2013
		US 2013-0212420 A1	15/08/2013
		US 9128472 B2	08/09/2015
US 2015-0006878 A1	01/01/2015	WO 2014-209922 A2	31/12/2014
WO 2015-094196 A1	25/06/2015	None	
US 2012-0124211 A1	17/05/2012	AU 2011-229697 A1	26/04/2012
		CA 2754304 A1	05/04/2012
		CN 102447743 A	09/05/2012
		EP 2439687 A1	11/04/2012
US 2012-0072597 A1	22/03/2012	US 2014-0317299 A1	23/10/2014
		US 8775626 B2	08/07/2014