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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
22 June 2006 (22.06.2006)

PCT

(10) International Publication Number
WO 2006/064018 A1

(51) International Patent Classification:
G06F 9/44 (2006.01) *G06F 9/46* (2006.01)

(21) International Application Number:
PCT/EP2005/056778

(22) International Filing Date:
14 December 2005 (14.12.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
11/013,884 15 December 2004 (15.12.2004) US

(71) Applicant *(for all designated States except US)*: **INTERNATIONAL BUSINESS MACHINES CORPORATION** [US/US]; New Orchard Road, Armonk, New York 10504 (US).

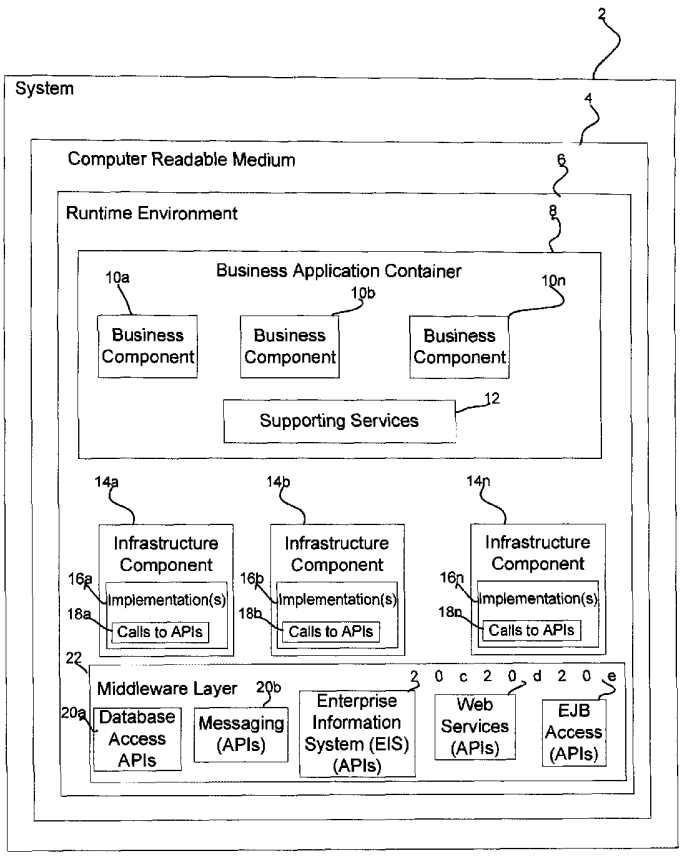
(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **ADAMS, Greg** [CA/CA]; 2996 Stoneridge Road, Dunrobin, Ontario KOA1TO (CA). **BEISIEGEL, Michael** [DE/US]; 21 Partners Trace, Poughkeepsie, NY 12603 (US). **BRODSKY, Stephen Andrew** [US/US]; 108 Mozart Avenue, Los Gatos, CA 95032 (US). **DELFINO, Jean-Sebastien Michel** [FR/US]; 2724 Brittan Avenue, San Carlos, CA 94070 (US). **FERGUSON, Donald** [US/US]; 2440 Pinetree Place, Yorktown Heights, NY 10598 (US). **HIGH, JR, Robert** [US/US]; 17 Wilderness Way, Round Rock, TX 78664 (US). **MCGEE, Jason Robert** [US/US]; 908 North York Court, Apex, NC 27502 (US). **NALLY, Martin Paul** [GB/US]; 1090 Katella Street, Laguna Beach, CA 92651 (US). **NIBLETT, Peter David** [GB/GB]; 1 Oakland Road, Whitchurch Hampshire RG28 7HH (GB). **SCHMIDT, Marc-Thomas** [DE/GB]; Grastead, Church Street, Upham, Southampton Hampshire SO22 5LG (GB).

(74) Agent: **ROBERTS, Scott**; IBM United Kingdom Limited, Intellectual Property Law, Hursley Park, Winchester Hampshire SO21 2JN (GB).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI,

[Continued on next page]

(54) Title: ACCESS MIDDLEWARE APPLICATION PROGRAMMING INTERFACES IN A RUNTIME ENVIRONMENT



(57) Abstract: Provided is an architecture for enabling business components to access middleware components in a runtime environment. A business container hosts business components and services to enable communication between the business components. A plurality of infrastructure components expose interfaces and methods to the business components, wherein the exposed interfaces and methods have names descriptive of a business domain for which the business applications are written. The infrastructure components implementation of the interfaces and methods exposed to the business components include calls to the middleware application programming interfaces (APIs) to invoke middleware APIs to cause middleware operations.

WO 2006/064018 A1

GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

ACCESS MIDDLEWARE APPLICATION PROGRAMMING INTERFACES IN A RUNTIME ENVIRONMENT

FIELD OF THE INVENTION

The present invention relates to an architecture for enabling business components to access middleware Application Programming Interfaces (APIs) in a runtime environment.

BACKGROUND OF THE INVENTION

Software developers often want to integrate business applications with various business services, such as web services, legacy applications, databases, Enterprise Information Systems (EIS), etc. One solution is the J2EE Connector Architecture, part of Java 2 Platform, Enterprise Edition (J2EE) 1.3, that specifies a standard architecture for accessing resources in diverse Enterprise Information Systems (EIS). The J2EE platform provides a reusable component model, using Enterprise JavaBeans and JavaServer Pages technologies to build and deploy multi-tier applications that are platform and vendor-independent. (Java, J2EE, Enterprise JavaBeans, and JavaServer Pages are trademarks of Sun Microsystems, Inc.).

An Enterprise Java Bean (EJB) is a collection of Java classes, following defined rules and providing specific call-back methods, and an XML file, combined into one single unit. Session beans model business services and expose EJB remote interfaces, which a client will use to invoke the services.

Oftentimes the developer of the business components or applications may not have detailed knowledge of the many application programming interfaces (APIs) needed to access the middleware layer components, including components such as database access components, messaging components, web service components, EJB, etc. Nonetheless, the developers of the business applications will have to spend considerable amounts of time to determine how to properly invoke the middleware layer from the business applications, which can be complex and technical.

SUMMARY OF THE INVENTION

Provided is an architecture for enabling business components to access middleware APIs in a runtime environment. A business container hosts business components and services to enable communication between the business components. A plurality of infrastructure components expose interfaces and methods to the business components, wherein the exposed interfaces and methods have names descriptive of a business domain for which the business applications are written. The infrastructure components implementation of the interfaces and methods exposed to the business components include calls to the middleware application programming interfaces (APIs) to invoke middleware APIs to cause middleware operations.

BRIEF DESCRIPTION OF THE DRAWINGS

A preferred embodiment of the present invention will now be described in detail by way of example only with reference to the following drawings:

Figure 1 illustrates an embodiment of a computing environment; and

Figures 2 and 3 illustrate operations to design components to implement business applications.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

In the following description, reference is made to the accompanying drawings which form a part hereof and which illustrate several embodiments of the present invention. It is understood that other embodiments may be utilized and structural and operational changes may be made.

Figure 1 illustrates a computing environment comprising a computer system 2 having a computer readable medium 4, comprising a volatile or non-volatile storage or memory. The computer readable medium 4 may comprise a memory device in which the described components are implemented and executed. The computer readable medium 4 includes a runtime environment 6 providing the overall runtime environment, such as the J2EE server, in which components and other containers run. The runtime environment 6 includes a business container 8 providing business components 10a, 10b…10n and supporting services 12 that allow the business components 10a, 10b…10n to run and call each other. The supporting services 12 may include life cycle management, security, deployment and component-specific services for the business components 10a, 10b…10n.

A plurality of infrastructure components 14a, 14b…14n each include one or more implementations 16a, 16b…16n, where each implementation 16a, 16b…16n includes calls to application programming interfaces (APIs) 18a, 18b…18n that are implemented in the middleware layer 22. The infrastructure components 14a, 14b…14n may expose interfaces to the business components 10a, 10b…10n that provide methods used to access the infrastructure component implementations 16a, 16b…16n. The implementations 16a, 16b…16n that may be invoked by the business components 10a, 10b…10n include application programming interfaces (APIs) 18a, 18b…18n to call the middleware APIs 20a, 20b, 20c, 20d, and 20e in a middleware layer 22. The middleware APIs may include: database access APIs 20a to provide access to a database; messaging APIs 20b, such as a Java Message Service (JMS), that allows communication with other entities; Enterprise Information System (EIS) APIs 20c that interface with EIS software and includes enterprise infrastructure systems such as enterprise resource planning (ERP), mainframe transaction processing, database systems, and other legacy information systems; web service APIs 20d providing access to services over the internet; and Enterprise JavaBeans (EJB) access APIs 20e. These APIs 20a, 20b, 20c, 20d, and 20e may be in their own containers. The variable "n" indicates an integer number of instances of an element, and may take different values when used with different elements, such that 10n, 14n, 16n, and 18n may indicate a same or different number of instances of the business components, infrastructure components, implementations, and calls to APIs, respectively.

In one embodiment, the business container 8 provides a runtime environment for the business components 10a, 10b…10n that operates within the general runtime environment 6. The business components 10a, 10b…10n include methods and themselves expose interfaces and methods to each other that have names, i.e., declarations descriptive of a business domain to which the business components are directed, such as financial services, retail operations, industrial operations, etc. The infrastructure components 14a, 14b…14n expose interfaces and methods to the business components 10a, 10b…10n that have names and declarations descriptive of the business domain of the business components 10a, 10b…10n that invoke the implementations 16a, 16b…16n of the infrastructure components 14a, 14b…14n. The implementations 16a, 16b…16n of the infrastructure components 14a, 14b…14n include calls to middleware APIs 18a, 18b…18n to invoke the middleware APIs 20a, 20b…20e on behalf of the business components 10a, 10b…10n. The middleware APIs 20a, 20b…20e may be included in middleware containers. In this way, those developing and coding the

business components 10a, 10b…10n may use methods and interfaces having names and declarations descriptive of the business domain in which they are operating to access the middleware components 20a, 20b…20n through the infrastructure components 14a, 14b…14n.

The developers of the business components 10a, 10b…10n need no knowledge of the APIs of the middleware components 20a, 20b…20e, which may be technical and complex, and need only focus on the business domain in which they are operating. For instance, to access data, such as a stock quote, the infrastructure component 14a, 14b…14n may expose an interface having a name descriptive of the business domain, e.g., getStockQuote(). However, the calls to the APIs 18a, 18b…18n in the implementations 16a, 16b…16n may invoke the specific technical APIs to access a database 20a or web service 20d to obtain the requested data. The developer of the business components 10a, 10b…10n does not need to be concerned with how the requested data is obtained, e.g., a database access 20a, web service 20c, etc., but only needs to use the interface whose name is descriptive of the operation as understood in the business domain, e.g., getStockQuote(). The developers of the infrastructure components 14a, 14b…14n have knowledge of how the middleware APIs 20a, 20b…20e may be used and invoked to provide the access of the middleware layer 22 on behalf of the business components 10a, 10b…10n.

Moreover, the business components 10a, 10b…10n may only call methods implemented in the business application container 8 and may not include any calls to services available in the runtime environment 6 outside the business application container 8. In this way, the business component developer only needs knowledge of the interfaces descriptive of the business domain and not the technical details of the runtime environment 6, such as the middleware APIs 20a, 20b…20e, which may be complex and require detailed knowledge of the different middleware components, such as J2EE and EJB, as well as the database and web service APIs.

Figure 2 illustrates operations performed by a developer of the components described in Figure 1. At block 100, the system oriented developer codes the infrastructure components 14a, 14b…14n to include one or more implementations 16a, 16b…16n having calls to the middleware APIs 18a, 18b…18n and to expose interfaces having names descriptive of the business domain to which the business components 10a, 10b…10n are directed. The interfaces exposed to the business components 10a, 10b…10n having names and declarations descriptive of the business domain invoke the implementations 16a, 16b…16n including the calls to the middleware

APIs 18a, 18b…18n to access the middleware functions. Each infrastructure component 14a, 14b…14n may perform a different one or more operations with respect to one of the middleware APIs 20a, 20b…20e. For instance, one middleware component may be used to access the database access API 20a and another infrastructure component one of the middleware components 20b, 20c…20e. Further, different infrastructure components 14a, 14b…14n may perform different operations with respect to the same middleware component 20a, 20b…20e. Yet further, one infrastructure component may call APIs with respect to different middleware applications 20a, 20b…20e.

Business level developers develop and code (at block 102) business components 10a, 10b…10n to run in the business application container 8 and include calls to the exposed interfaces, descriptive of the business domain, of the infrastructure components 14a, 14b…14n to access data (or access services of middleware outside of the business container 10). The business components 10a, 10b…10n may only include calls to services within the business container 8 and interfaces exposed by the infrastructure components 10a, 10b…10n. The developed infrastructure components 14a, 14b…14n and business application container 8 including business components 10a, 10b…10n are deployed into the runtime environment 8.

In certain situations, the developer may want to change the middleware API 20a, 20b…20e used to perform operations. For instance, the developer may want a business component 10a, 10b…10n to access data from the web service API 20d instead of the database access API 20a. At block 150, the developer may decide to use different middleware APIs 20a, 20b…20e to access data requested by the business components 10a, 10b…10n. To accomplish this, those working on the infrastructure components 14a, 14b…14n, may code (at block 152) infrastructure components 14a, 14b…14n to maintain the same named interfaces exposed to the business components 10a, 10b…10n, descriptive of the business domain, but create new infrastructure components 14a, 14b…14n or modify the existing infrastructure components 14a, 14b…14n to include calls to the new middleware APIs that will be used to access the data (or access middleware APIs outside of the business application container 8). The newly coded or modified infrastructure components 14a,14b…14n may then be deployed (at block 154) in the runtime environment 6 to be available to calls from the business components 10a, 10b…10n and to invoke the new (second) set of middleware APIs 20a, 20b…20n to perform operations previously performed by a previous (first set) of middleware APIs 20a, 20b…20e, such as access data or perform other middleware operations.

The described embodiments thus provide a way to separate the types of operations and calls into different layers, such as a business layer and infrastructure layer, so that developers of the business components need not be concerned with the details of how the calls, which are descriptive of the business domain, to the infrastructure components to obtain data are implemented.

The described operations may be implemented as a method, apparatus or article of manufacture using standard programming and/or engineering techniques to produce software, firmware, hardware, or any combination thereof. The term "article of manufacture" as used herein refers to code or logic implemented in hardware logic (e.g., an integrated circuit chip, Programmable Gate Array (PGA), Application Specific Integrated Circuit (ASIC), etc.) or a computer readable medium, such as magnetic storage medium (e.g., hard disk drives, floppy disks,, tape, etc.), optical storage (CD-ROMs, optical disks, etc.), volatile and non-volatile memory devices (e.g., EEPROMs, ROMs, PROMs, RAMs, DRAMs, SRAMs, firmware, programmable logic, etc.). Code in the computer readable medium is accessed and executed by a processor. The code in which preferred embodiments are implemented may further be accessible through a transmission media or from a file server over a network. In such cases, the article of manufacture in which the code is implemented may comprise a transmission media, such as a network transmission line, wireless transmission media, signals propagating through space, radio waves, infrared signals, etc. Thus, the "article of manufacture" may comprise the medium in which the code is embodied. Additionally, the "article of manufacture" may comprise a combination of hardware and software components in which the code is embodied, processed, and executed. Of course, those skilled in the art will recognize that many modifications may be made to this configuration, and that the article of manufacture may comprise any information bearing medium known in the art.

The illustrated operations of Figures 2 and 3 show certain events occurring in a certain order. In alternative embodiments, certain operations may be performed in a different order, modified or removed. Moreover, steps may be added to the above described logic and still conform to the described embodiments. Further, operations described herein may occur sequentially or certain operations may be processed in parallel. Yet further, operations may be performed by a single processing unit or by distributed processing units.

CLAIMS

1. A method, comprising:

providing a business container hosting business components and services to enable communication between the business components;

providing a plurality of infrastructure components, wherein the infrastructure components expose interfaces and methods to the business components, wherein the exposed interfaces and methods have names descriptive of a business domain for which the business applications are written; and

providing middleware application programming interfaces (APIs), wherein the infrastructure components implementation of the interfaces and methods exposed to the business components include calls to the middleware APIs to invoke middleware APIs to cause middleware operations.

2. The method of claim 1, further comprising:

providing at least one middleware container, wherein at least one middleware API invoked by the infrastructure components executes in the middleware container.

3. The method of claim 1, wherein the business components access the middleware APIs through the infrastructure components, and wherein the business components do not include any direct calls to the middleware APIs.

4. The method of claim 1, wherein the business container, infrastructure components, and middleware APIs all execute in a runtime environment, and wherein the business container provides a runtime environment for the business components.

5. The method of claim 4, wherein the business components do not directly call any services and interfaces outside of the business container, wherein the business components call infrastructure components, and wherein the called infrastructure components call middleware APIs and other services in the runtime environment external to the business container.

6. The method of claim 1, wherein the business components include calls to the infrastructure components to access data from a first middleware API, further comprising:

selecting a second middleware API from which to access the data requested by the business component;

coding the infrastructure components to include calls to the selected second middleware APIs to access the data on behalf of the business components previously accessed through the first middleware API, wherein the names of the interfaces used to invoke the infrastructure components to call the second middleware API are the names of the interfaces use to call the infrastructure component to interact with the first middleware API.

7. The method of claim 1, wherein the business components include calls to the infrastructure components to access a first set of middleware APIs external to the business container, further comprising:

selecting a second set of middleware APIs to perform operations for the business components that are currently performed by a first set of middleware APIs; and

coding the infrastructure components to include calls to the second set of middleware APIs to perform the operations requested by the business components previously performed by the first set of middleware APIs, wherein the names of the interfaces called by the business components to invoke the infrastructure components to call the second set of middleware APIs are the same names used to invoke the infrastructure components to call the first set of middleware APIs.

8. A system, comprising:

a processor;

a computer readable medium accessible to the processor and including:

(i) a business container hosting business components and services to enable communication between the business components;

(ii) a plurality of infrastructure components, wherein the infrastructure components expose interfaces and methods to the business components, wherein the exposed interfaces and methods have names descriptive of a business domain for which the business applications are written; and

(iii) middleware application programming interfaces (APIs), wherein the infrastructure components implementation of the interfaces and methods exposed to the business components include calls to the middleware APIs to invoke middleware APIs to cause middleware operations.

9. The system of claim 8, wherein the computer readable medium further includes:

at least one middleware container, wherein at least one middleware API invoked by the infrastructure components executes in the middleware container.

10. The system of claim 8, wherein the business components access the middleware APIs through the infrastructure components, and wherein the business components do not include any direct calls to the middleware APIs.

11. The system of claim 8, wherein the business container, infrastructure components, and middleware APIs all execute in a runtime environment executed by the processor, and wherein the business container provides a runtime environment executed by the processor for the business components.

12. The system of claim 11, wherein the business components do not directly call any services and interfaces outside of the business container, wherein the business components call infrastructure components, and wherein the called infrastructure components call middleware APIs and other services in the runtime environment external to the business container.

13. The system of claim 8, wherein the business components include calls to the infrastructure components to access data from a first middleware API, wherein a developer services the code by performing operations comprising:

selecting a second middleware API from which to access the data requested by the business component;

coding the infrastructure components to include calls to the selected second middleware APIs to access the data on behalf of the business components previously accessed through the first middleware API, wherein the names of the interfaces used to invoke the infrastructure components to call the second middleware API are the names of the interfaces use to call the infrastructure component to interact with the first middleware API.

14. The system of claim 8, wherein the business components include calls to the infrastructure components to access a first set of middleware APIs external to the business container, wherein a developer services the code by performing operations comprising:

selecting a second set of middleware APIs to perform operations for the business components that are currently performed by a first set of middleware APIs; and

coding the infrastructure components to include calls to the second set of middleware APIs to perform the operations requested by the business components previously performed by the first set of middleware APIs, wherein the names of the interfaces called by the business components to invoke the infrastructure components to call the second set of middleware APIs are the same names used to invoke the infrastructure components to call the first set of middleware APIs.

15. An article of manufacture including code capable of causing operations to be performed, the operations comprising:

providing a business container hosting business components and services to enable communication between the business components;

providing a plurality of infrastructure components, wherein the infrastructure components expose interfaces and methods to the business components, wherein the exposed interfaces and methods have names descriptive of a business domain for which the business applications are written; and

providing middleware application programming interfaces (APIs), wherein the infrastructure components implementation of the interfaces and methods exposed to the business components include calls to the middleware APIs to invoke middleware APIs to cause middleware operations.

16. The article of manufacture of claim 15, wherein the operations further comprise:

providing at least one middleware container, wherein at least one middleware API invoked by the infrastructure components executes in the middleware container.

17. The article of manufacture of claim 15, wherein the business components access the middleware APIs through the infrastructure components, and wherein the business components do not include any direct calls to the middleware APIs.

18. The article of manufacture of claim 15, wherein the business container, infrastructure components, and middleware APIs all execute in a runtime environment, and wherein the business container provides a runtime environment for the business components.

19. The article of manufacture of claim 18, wherein the business components do not directly call any services and interfaces outside of the business container, wherein the business components call infrastructure components, and wherein the called infrastructure components call middleware APIs and other services in the runtime environment external to the business container.

20. The article of manufacture of claim 15, wherein the business components include calls to the infrastructure components to access data from a first middleware API, wherein a developer services the code by performing operations comprising:

selecting a second middleware API from which to access the data requested by the business component;

coding the infrastructure components to include calls to the selected second middleware APIs to access the data on behalf of the business components previously accessed through the first middleware API, wherein the names of the interfaces used to invoke the infrastructure components to call the second middleware API are the names of the interfaces use to call the infrastructure component to interact with the first middleware API.

21. The article of manufacture of claim 15, wherein the business components include calls to the infrastructure components to access a first set of middleware APIs external to the business container, wherein a developer services the code by performing operations comprising:

selecting a second set of middleware APIs to perform operations for the business components that are currently performed by a first set of middleware APIs; and

coding the infrastructure components to include calls to the second set of middleware APIs to perform the operations requested by the business components previously performed by the first set of middleware APIs, wherein the names of the interfaces called by the business components to invoke the infrastructure components to call the second set of middleware APIs are the same names used to invoke the infrastructure components to call the first set of middleware APIs.

FIG. 1

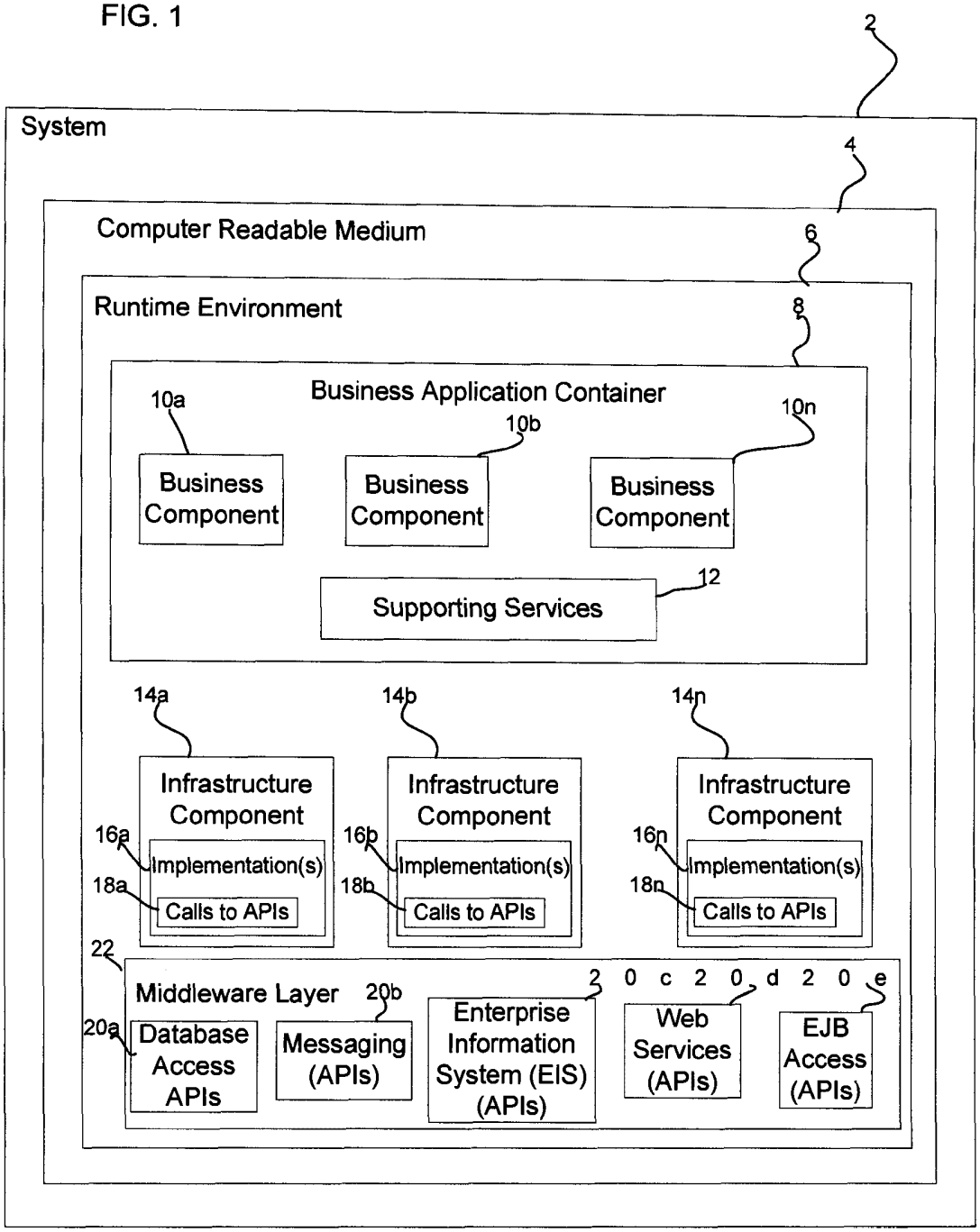

2
System
4
Computer Readable Medium
6
Runtime Environment
8
Business Application Container
10a
10b
10n
Business Component
Business Component
Business Component
12
Supporting Services
14a
14b
14n
Infrastructure Component
Infrastructure Component
Infrastructure Component
16a
16b
16n
Implementation(s)
Implementation(s)
Implementation(s)
18a
18b
18n
Calls to APIs
Calls to APIs
Calls to APIs
22
Middleware Layer
20b
20a
Database Access APIs
Messaging (APIs)
Enterprise Information System (EIS) (APIs)
Web Services (APIs)
EJB Access (APIs)

FIG. 2

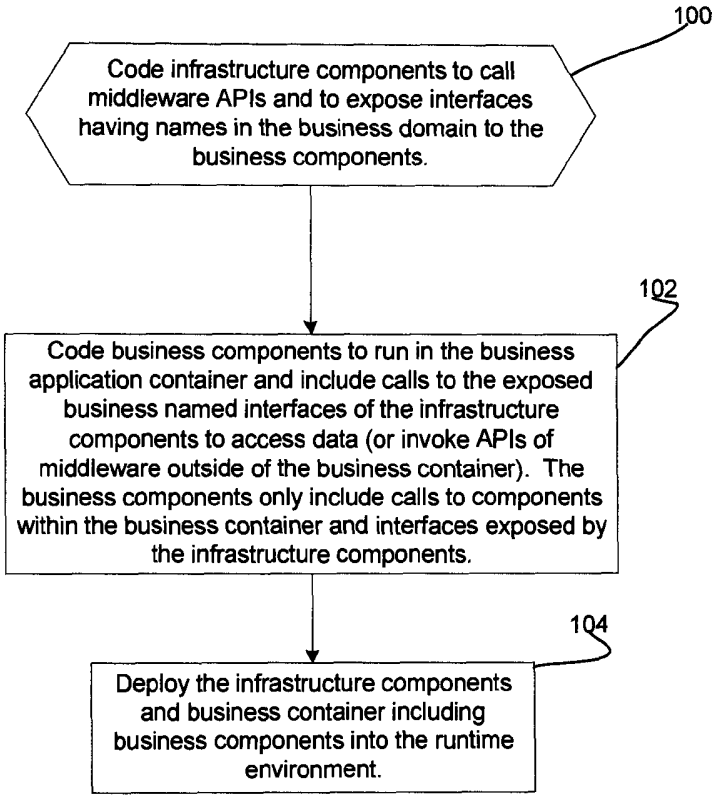
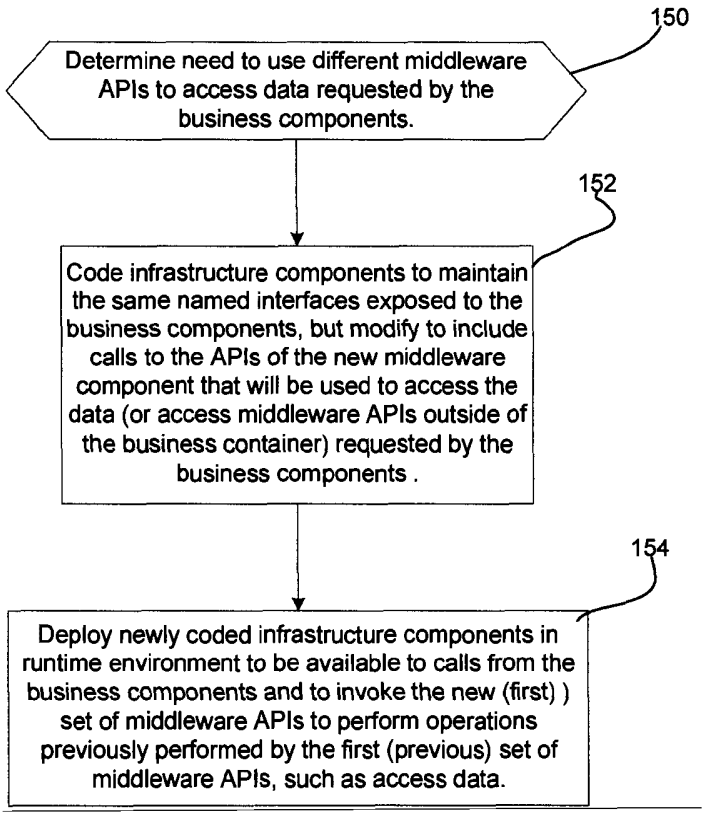

100
Code infrastructure components to call middleware APIs and to expose interfaces having names in the business domain to the business components.
102
Code business components to run in the business application container and include calls to the exposed business named interfaces of the infrastructure components to access data (or invoke APIs of middleware outside of the business container). The business components only include calls to components within the business container and interfaces exposed by the infrastructure components.
104
Deploy the infrastructure components and business container including business components into the runtime environment.

FIG. 3


150
Determine need to use different middleware APIs to access data requested by the business components.
152
Code infrastructure components to maintain the same named interfaces exposed to the business components, but modify to include calls to the APIs of the new middleware component that will be used to access the data (or access middleware APIs outside of the business container) requested by the business components .
154
Deploy newly coded infrastructure components in runtime environment to be available to calls from the business components and to invoke the new (first)) set of middleware APIs to perform operations previously performed by the first (previous) set of middleware APIs, such as access data.

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2005/056778

A. CLASSIFICATION OF SUBJECT MATTER
G06F9/44 G06F9/46

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)
G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, IBM-TDB

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2004/243693 A1 (BEISIEGEL MICHAEL ET AL) 2 December 2004 (2004-12-02) abstract paragraphs [0001] - [0003] paragraphs [0007] - [0013] paragraphs [0029], [0030] paragraph [0035] paragraph [0042] claims 1-5 ----- -/--	1-21

[X] Further documents are listed in the continuation of Box C.

[X] 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 document 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
24 March 2006	31/03/2006

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Tx. 31 651 epo nl, Fax: (+31–70) 340–3016	Tomàs Blanch, F

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2005/056778

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2003/182347 A1 (DEHLINGER PATRICK) 25 September 2003 (2003-09-25) abstract paragraphs [0003], [0004] paragraphs [0010] - [0015] paragraphs [0024] - [0031] paragraphs [0078] - [0096] paragraph [0104] claim 1	1-21
X	WO 03/069472 A (TELEFONAKTIEBOLAGET LM ERICSSON ; WILHELMSSON, LARS; BJAERE, BJOERN; H) 21 August 2003 (2003-08-21) abstract page 3, line 10 - page 4, line 8 page 5, line 20 - page 6, line 16 page 9, lines 20-25 claims 1,2	1-21
A	US 2003/195997 A1 (IBERT TERENCE WINFIELD ET AL) 16 October 2003 (2003-10-16) the whole document	1-21
A	US 2003/188043 A1 (WOODALL THOMAS R ET AL) 2 October 2003 (2003-10-02) the whole document	1-21

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/EP2005/056778

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2004243693 A1	02-12-2004	CA 2357168 A1	10-03-2003
		WO 03024054 A2	20-03-2003
		TW 582147 B	01-04-2004
US 2003182347 A1	25-09-2003	AT 235081 T	15-04-2003
		AU 9004401 A	02-04-2002
		DE 60100128 D1	24-04-2003
		EP 1191437 A1	27-03-2002
		FR 2814257 A1	22-03-2002
		WO 0225437 A1	28-03-2002
WO 03069472 A	21-08-2003	AU 2003214058 A1	04-09-2003
		CN 1633641 A	29-06-2005
		EP 1476809 A2	17-11-2004
		JP 2005518015 T	16-06-2005
		US 2003181193 A1	25-09-2003
US 2003195997 A1	16-10-2003	NONE	
US 2003188043 A1	02-10-2003	NONE	