



US007356771B2

(12) **United States Patent**
Paul et al.

(10) **Patent No.:** **US 7,356,771 B2**
(45) **Date of Patent:** **Apr. 8, 2008**

(54) **ADAPTIVE CONTENT PLATFORM AND
METHOD OF USING SAME**

(75) Inventors: **Santanu Paul**, Cambridge, MA (US);
Gary Zakon, Lexington, MA (US)
(73) Assignee: **Openpages**, Waltham, MA (US)
(*) Notice: Subject to any disclaimer, the term of this
patent is extended or adjusted under 35
U.S.C. 154(b) by 602 days.

(21) Appl. No.: **10/256,613**

(22) Filed: **Sep. 26, 2002**

(65) **Prior Publication Data**

US 2004/0010796 A1 Jan. 15, 2004

Related U.S. Application Data

(60) Provisional application No. 60/394,441, filed on Jul.
9, 2002.

(51) **Int. Cl.**
G06F 3/00 (2006.01)
G06F 7/00 (2006.01)
G06F 9/44 (2006.01)

(52) **U.S. Cl.** **715/744**; 707/3; 715/762;
717/100

(58) **Field of Classification Search** 719/311,
719/328–332, 318; 709/201–207, 217–219;
707/1–10, 102, 201–206; 715/501.1, 733,
715/734, 738, 740–753, 513, 762, 763; 726/1–21,
726/27–30; 705/1–9; 717/100–105

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,401,094 B1 * 6/2002 Stemp et al. 707/10
6,816,904 B1 * 11/2004 Ludwig et al. 709/226
2002/0083124 A1 * 6/2002 Knox et al. 709/203
2002/0152286 A1 * 10/2002 Peddu et al. 709/219
2003/0023953 A1 * 1/2003 Lucassen et al. 717/106
2003/0046201 A1 * 3/2003 Cheyer 705/35
2003/0105644 A1 * 6/2003 Ali et al. 705/1
2003/0222906 A1 * 12/2003 Fish et al. 345/744
2006/0074727 A1 * 4/2006 Briere 705/8

* cited by examiner

Primary Examiner—William Thomson

Assistant Examiner—Charles E Anya

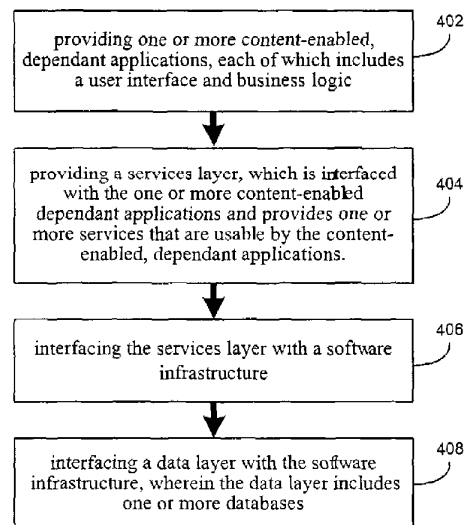
(74) *Attorney, Agent, or Firm*—Wilmer Cutler Pickering
Hale & Dorr LLP

(57) **ABSTRACT**

An adaptive content platform includes one or more content-enabled, dependent applications, each of which includes a user interface and business logic. A services layer, which is interfaced with the dependent applications and a software infrastructure, provides one or more services that are usable by the dependent applications.

146 Claims, 8 Drawing Sheets

400



Google Exhibit 1040
Google v. VirtaMove

Oracle Ex. 1040
Oracle v. VirtaMove, IPR2025-00965

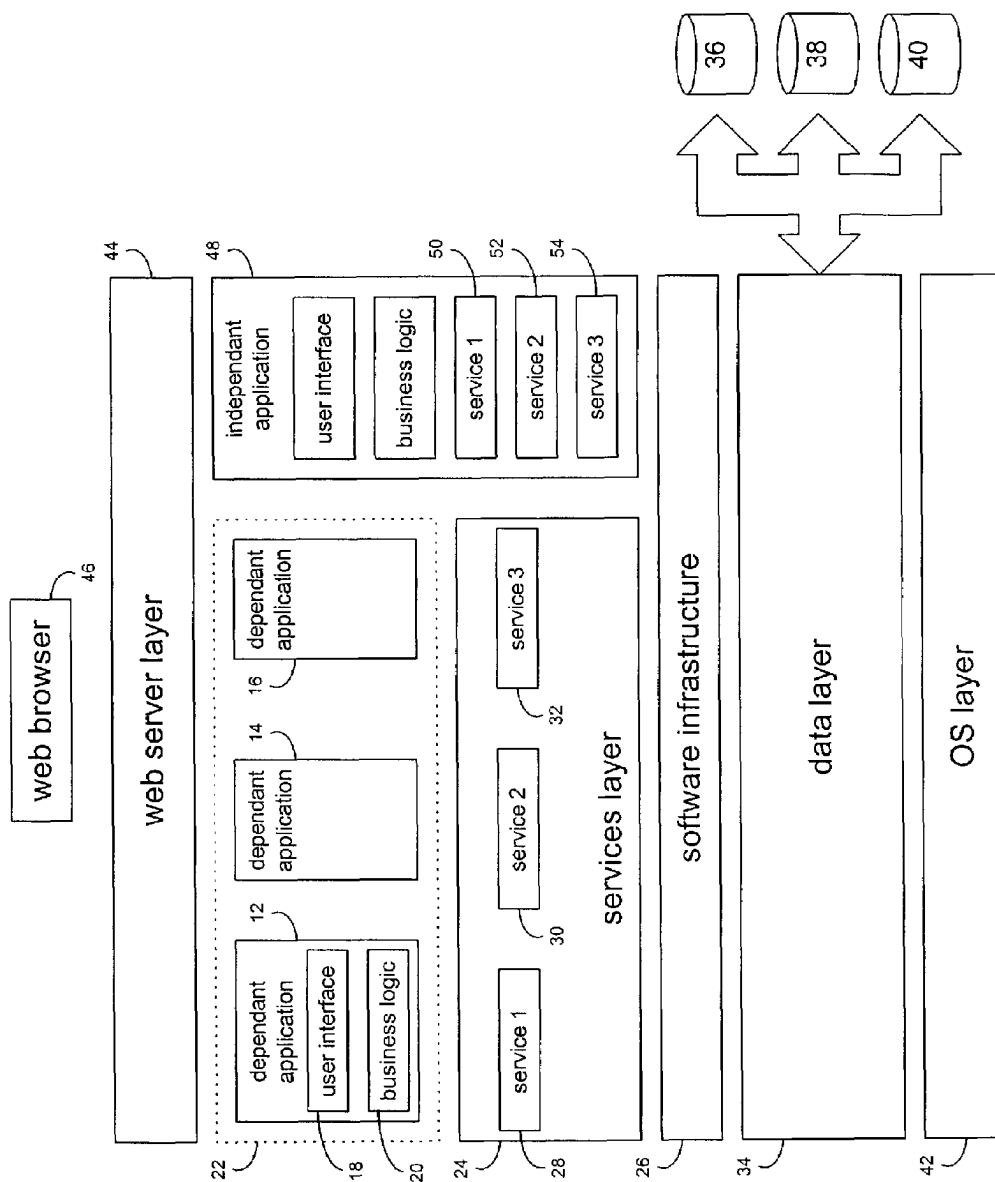


Fig. 1

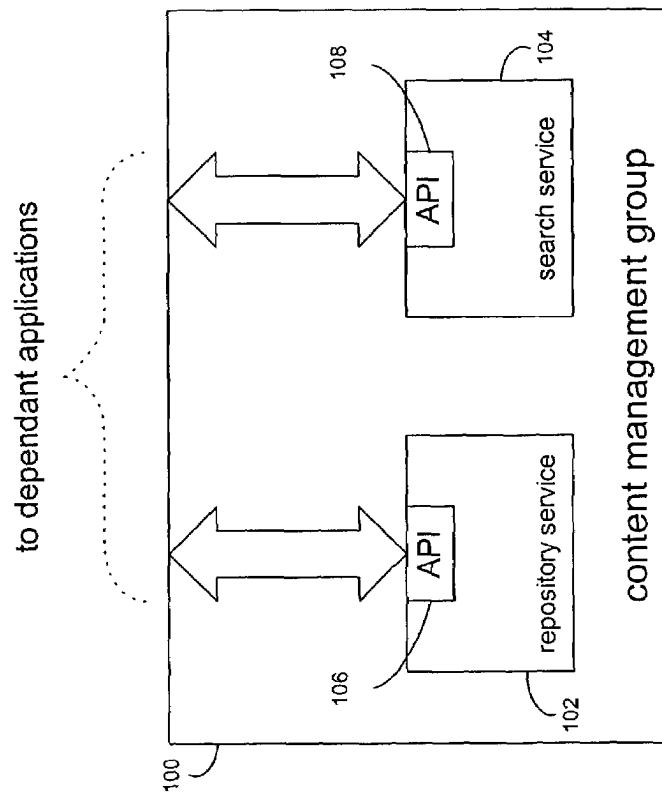


Fig. 2

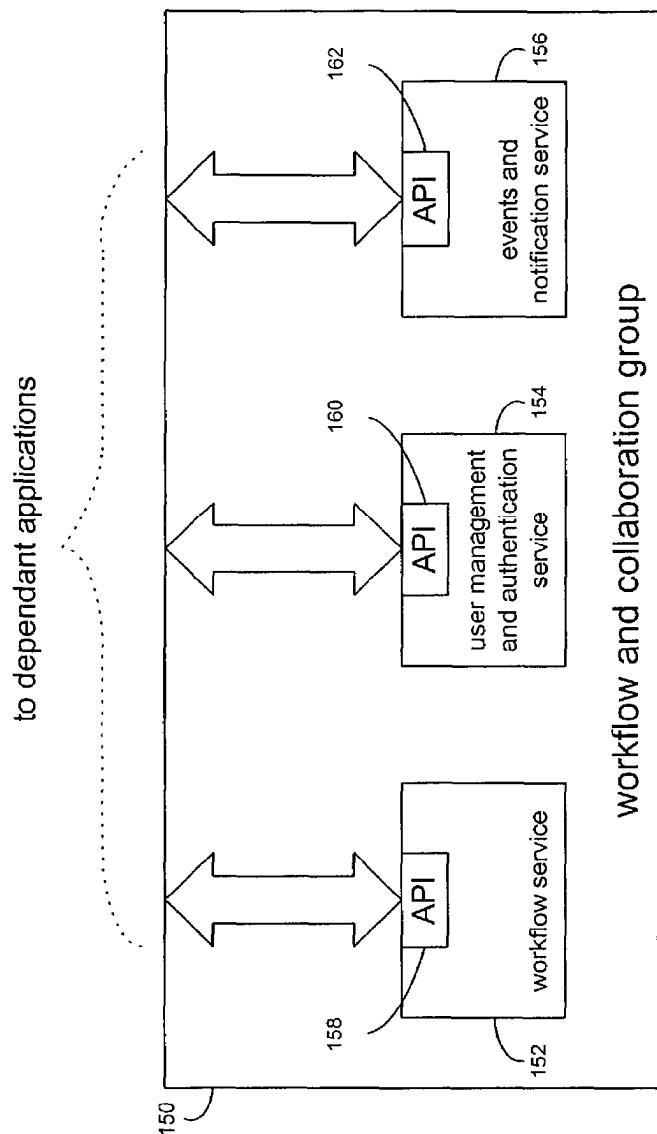


Fig. 3

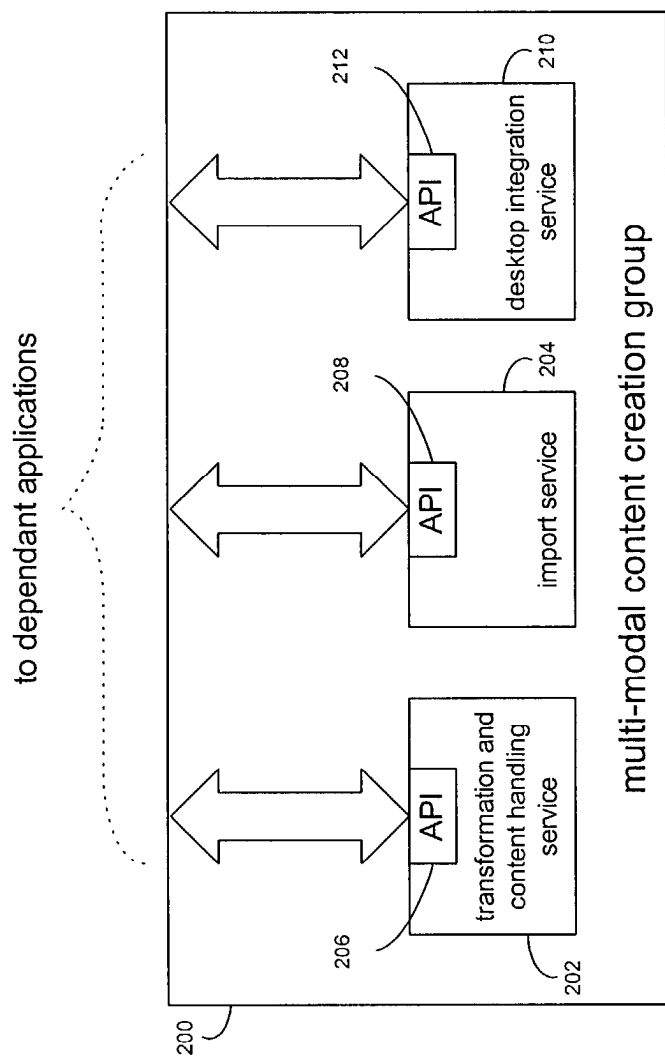


Fig. 4

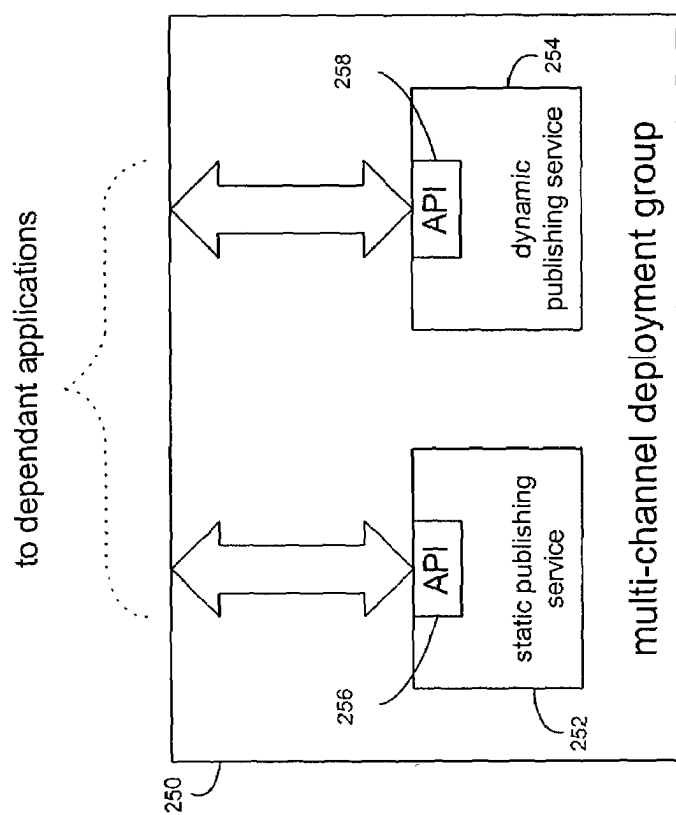


Fig. 5

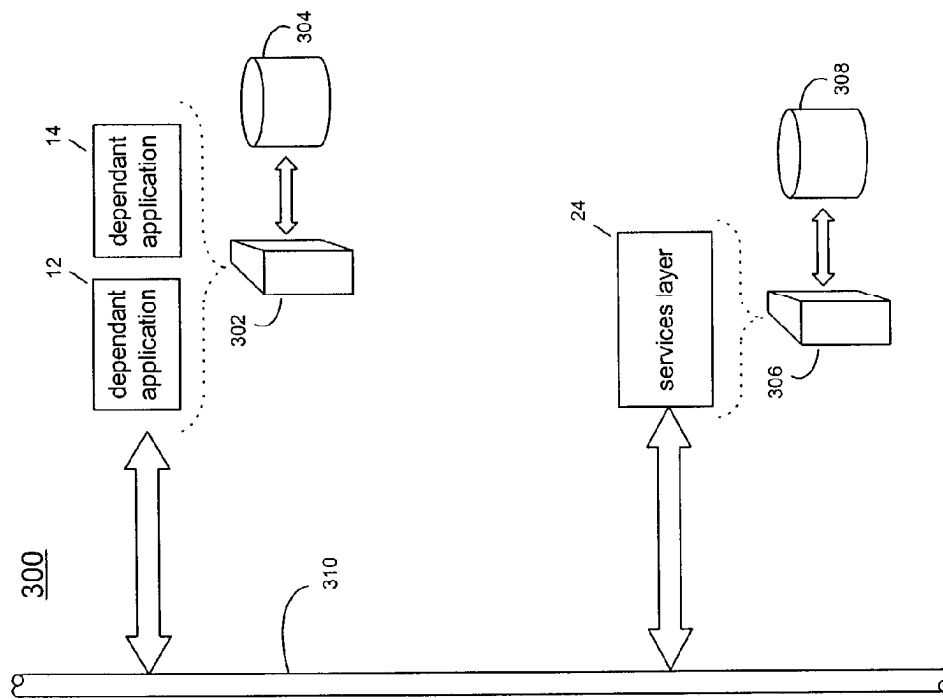


Fig. 6

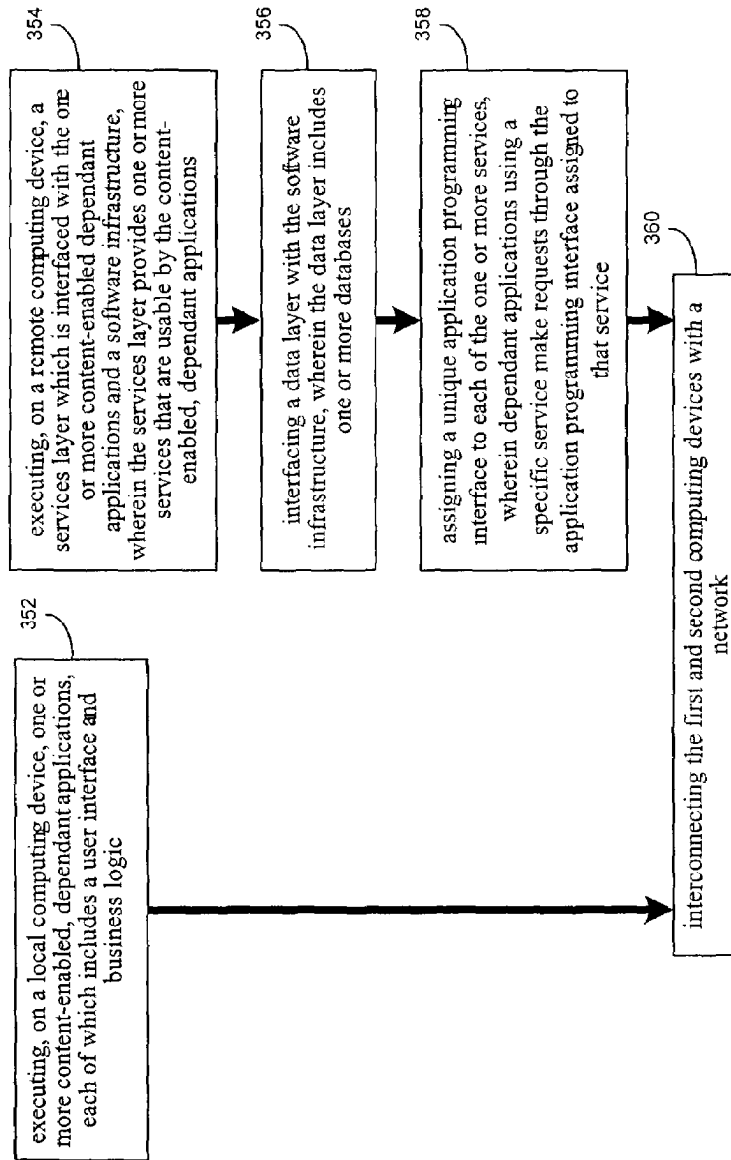


Fig. 7

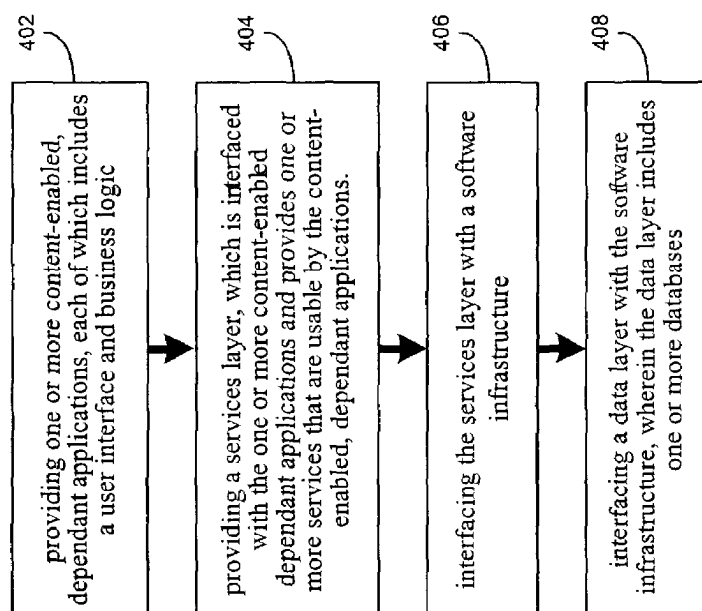
400

Fig. 8

1

ADAPTIVE CONTENT PLATFORM AND METHOD OF USING SAME

RELATED APPLICATIONS

This application claims the priority of: U.S. Provisional Patent Application No. 60/394,441, entitled "Adaptive Content Platform", and filed on Jul. 9, 2002.

BACKGROUND

This invention relates to software development, and more particularly to software development platforms.

For a suite of applications developed for a common software platform (e.g., an application server platform), each application within the suite typically includes a group of services (e.g., content management services, workflow services, publishing services, search and query services) that are common amongst the applications in the suite.

As the software platform does not allow the services of one application to be shared by another, these services must be coded for and included in each application written for the software platform, even though multiple applications use common services.

SUMMARY

According to an aspect of this invention, an adaptive content platform includes one or more content-enabled, dependent applications (thus forming an application layer), each of which includes a user interface and business logic. A services layer, which is interfaced with the content-enabled dependent applications and a software infrastructure (e.g., an application server), provides one or more services that are usable by the content-enabled, dependent applications.

One or more of the following features may be included. The services layer includes a unique application programming interface for each of the one or more services, such that dependent applications using a specific service make requests through the application programming interface assigned to that service.

The services include a content management service (for storing and managing files), which includes: a repository service for storing files; and a search service for allowing users to search files stored by the repository service, for example.

The services also include a workflow and collaboration service (for managing projects and users), which includes: a workflow service for managing the workflow of files; a user management and authentication service for managing the users and user groups; and an events and notification service for managing and broadcasting notifications to the users that were generated by the services, for example.

Additionally, the services include a multi-modal content creation service (for manual creation and automated importation and conversion of files), which includes: a transformation and content handling service for converting the formats of files; a desktop integration service for manual contribution of content; and an import service for facilitating file importation from external systems, for example.

The services further include a multi-channel deployment service (for publishing files to one or more publishing channels), which includes: a publishing service for publishing files to one or more publishing channels (such as a web site, an email broadcast, a wireless broadcast, a syndication

2

stream, or a printed publication, for example); a dynamic publishing service; and a static publishing service, for example.

A data layer is interfaced with the software infrastructure, such that the data layer includes one or more databases.

The above-described adaptive content platform may be implemented in a distributed computing system, in that a first (or local) computing device executes the content-enabled, dependent applications, and a second (or remote) computing device executes the services layer. This second computer may be a web server.

In a distributed computing system, the first and second computing devices are interconnected by a network, such as a local area network, the Internet, or an intranet, for example.

The above-described adaptive content platform and distributed computing system may be implemented as a method or a sequence of instructions executed by a processor.

One or more advantages can be provided from the above. By providing the programmer with a common set of reusable services, applications are no longer required to include stand-alone services. Since these applications are not required to include services, application development is significantly accelerated and deployment is simplified. Additionally, by using a common set of services, compatibility issues are minimized. Further, as applications share a common set of services, application size is reduced. In addition, by separating an application's services from the application's business logic/user interface, distributed computing is possible, leading to superior scalability and availability.

The details of one or more embodiments of the invention are set forth in the accompanying drawings and the description below. Other features, objects, and advantages of the invention will be apparent from the description and drawings, and from the claims.

DESCRIPTION OF DRAWINGS

FIG. 1 is a block diagram of an adaptive content platform;

FIG. 2 is a block diagram of a content management group of the adaptive content platform;

FIG. 3 is a block diagram of a workflow and collaboration group of the adaptive content platform;

FIG. 4 is a block diagram of a multi-modal content creation group of the adaptive content platform;

FIG. 5 is a block diagram of a multi-channel deployment group of the adaptive content platform;

FIG. 6 is a block diagram the adaptive content platform incorporated into a distributed computing system;

FIG. 7 is a flow chart of a distributed computing method; and

FIG. 8 is a flow chart of a multi-tier software development method.

DETAILED DESCRIPTION

Referring to FIG. 1, an adaptive content platform 10 includes one or more dependent applications 12, 14, 16, each of which includes a user interface (e.g., user interface 18) and business logic (e.g., business logic 20). This group of dependent applications may be referred to as an application layer 22. Each user interface 18 allows a user of the dependent application to access and use the functionality of the dependent application. Business logic 20 performs the functions native to the dependent application. Note that while dependent applications typically include some form of user interface, this is not required.

Typically, the dependent applications **12**, **14**, **16**, are content-enabled dependent applications, in that they manage and process content (e.g., documents, images, audio clips, video clips). Examples of content-enabled applications are web content management systems, shareholder and regulatory reporting applications, corporate marketing and communications systems, newspaper publishing systems, and so forth. Content-enabled applications typically include a combination of content management, workflow management, and publishing capabilities.

Adaptive content platform **10** is a multi-tiered software architecture that includes a services layer **24** for interfacing the application layer **22** and a software infrastructure **26** (e.g., an application server). Examples of an application server are BEA Weblogic™, and IBM Websphere™, both of which implement the Java™ 2 Enterprise Edition standard (J2EE).

Services layer **24** provides a group of services **28**, **30**, **32**, which are available for use by dependent applications **12**, **14**, **16**. Examples of these services include content management services, search services, and file conversion services, for example. These services, which will be discussed below in greater detail, are shared services common to the dependent applications; as opposed to each dependent application's native functions, which are handled by the dependent application's business logic.

Dependent applications **12**, **14**, **16** maybe J2EE ("Java™ 2 Enterprise Edition") compliant dependent applications that adhere to v1.3 standards and are compatible with and run on a Java™ 2 Enterprise Edition application server.

A data layer **34** is interfaced to the software infrastructure **26**, and provides data services for platform **10**. Data layer **34** may provide access to database servers **36**, such as Oracle™, IBM DB2™, and Microsoft SQL Server™. Further, data layer **34** may provide access to file servers **38**, such as Microsoft Windows 2000 Servers™, Microsoft Windows NT Servers™, and Unix Servers™. Additionally, data layer **34** may allow access to legacy systems **40** (i.e., applications and data that have been inherited from languages, platforms, and techniques earlier than current technology).

Data layer **34** is typically interfaced with an operating system (OS) layer **42**, which includes the operating system that manages the above-described layers, infrastructures, and dependent applications. Examples of compatible operating systems are Windows™, Solaris™, and Linux™.

Typically, a web server layer **44** is interfaced with the application layer (i.e., dependent applications **12**, **14**, **16**) and allows a user (not shown) to use and access the functionality of the individual dependent applications from with a web browser **46** (e.g., Microsoft Internet Explorer™, Netscape Navigator™). Examples of web server layer **44** are Microsoft Internet Information Server™, and Apache web server™.

By combining the user interface **18** and business logic **20** of a dependent application **12** with one or more of the services **28**, **30**, **32** offered by the services layer **24**, the functionality of a "stand-alone" independent application **48** can be emulated without the application having to include dedicated services **50**, **52**, **54**.

Concerning the services **28**, **30**, **32**, offered by services layer **24**, these services can typically be loosely described as four groups of services, namely: content management; workflow and collaboration; multi-modal content creation; and multi-channel deployment, each of which will be discussed below in greater detail.

Referring to FIG. 2, the content management group **100**, which stores and manages files and content used by the

adaptive content platform, may include a repository service **102** and a search service **104**.

Repository service **102** works in conjunction with the data layer (generally), and the database servers, the file servers, and the legacy systems (specifically) to store, organize, and manage files and content (hereinafter "files").

Repository service **102** allows for the production, organization, and management of numerous content types that define the specific type of files being produced and managed. Additionally, repository service **102** allows users/administrators to define numerous property fields or meta-data fields (e.g., release date, revision number, production date, revision date, and approval date, for example) that define and refine the files stored by the data layer. Access to the files managed by repository service **102** can be controlled by regulating the users who can view, check out, edit, print, and save a particular file, for example. Additionally, the data structure in which the files are stored (e.g., the directory tree structure) is defined and controlled using repository service **102**. Typically, repository service **102** works in conjunction with a relational database, e.g., database **36** (FIG. 1), that is accessed through data layer **34** (FIG. 1).

The search service **104** allows a user to search the files stored by the repository service **102**. Searches may be performed on either file properties or content. If the files are stored in a structured database (as described above), search service **104** may be an SQL ("structured query language") database query engine. Alternatively, if the files are stored as HTML or XML ("Extensible Markup Language") based documents, search service **104** may use search engine technology to generate a list of relevant documents.

The dependent applications **12**, **14**, **16** (FIG. 1) described above may access each service offered by content management group **100** (e.g., repository service **102** and search service **104**) by making the appropriate request of and establishing a connection through the API ("application programming interface") assigned to that particular service. For example, API **106** is assigned to repository service **102**, and API **108** is assigned to search service **104**. Therefore, if a user of a dependent application (e.g., dependent application **12**, FIG. 1) wanted to execute a search for a particular file, dependent application **12** would make the appropriate request from API **108**.

Referring to FIG. 3, the workflow and collaboration group **150**, which manages projects and users of the adaptive content platform, may include a workflow service **152**, a user management and authentication service **154**, and an events and notification service **156**.

The workflow service **152** allows the administrator or user to control the workflow of files through the adaptive content platform. For example, if a file is produced for publishing purposes, that file might need to be approved by a midlevel manager prior to it being sent to an upper level manager. Further, the upper level manager might have to approve the file prior to it being published or otherwise disseminated. Therefore, workflow service **152** could mandate that the file be approved by a midlevel manager prior to it being sent to the higher level manager, who approves it prior to publication. Further, workflow service **152** may assign time limits for the completion of certain tasks, such as the midlevel or upper level review and approval process.

The user management and authentication service **154** provides a set of tools to the user/administrator that allows them to manage users and user groups. Individual users can be produced and deleted using user management and authentication service **154**. Further, the rights and privileges of these individual users also can be controlled and regulated.

Additionally, these users can be assigned to, moved between, and deleted from various users groups, which are also maintained using user management and authentication service **154**. Further, as rights and privileges can be assigned to a user group, by adding an individual user to a user group, the rights or privileges of an individual user can be efficiently defined.

The events and notification service **156** allows for the delivery of notification events generated by the services offered by the applet service layer **24** (FIG. 1). These message can be delivered to individual users of the system, broadcast to entire user groups, or delivered to the various services offered by the applet service layer **24** (FIG. 1).

As above, the dependent applications **12, 14, 16** (FIG. 1) described above may access each service offered by workflow and collaboration group **150** (e.g., workflow service **152**, user management and authentication service **154**, and the events and notification service **156**) by making the appropriate request of and establishing a connection through the API assigned to that particular service. For this particular group, API **158** is assigned to the workflow service **152**, API **160** is assigned to the user management and authentication service **154**, and API **162** is assigned to the events and notification service **156**.

Referring to FIG. 4, the multi-modal content creation group **200**, which imports and converts files for the adaptive content platform, may include a transformation and content handling service **202**, an import service **204**, and a desktop integration service **210**.

The transformation and content handling service **202** provides file format conversion services, thus allowing the user to import files of various types and convert them over into a common format (e.g., XML and HTML). Converter templates are available for popular applications, such as Microsoft Word™, Microsoft Excel™, Adobe PDF™, and Microsoft PowerPoint™, for example.

The import service **204** allows for automated import of files from external systems. Import service **204** is configured to monitor, on a periodic basis, the files located on a network drive, an FTP ("file transfer protocol") site, and an HTTP site. When new files are detected on one of these sources, the files are automatically imported into the system. Further, if a format conversion is required, import service **204** will work in conjunction with transformation service **202** to import and convert the file.

The desktop integration service **210** allows content to be contributed by users via standard desktop creation tools. These tools include the Microsoft Office™ suite, as well as Adobe™ and Macromedia™ applications. The service uses the WEBDAV protocol (WEB-based Distributed Authoring and Versioning), which is an extension of the HTTP protocol, to communicate with the desktop tools.

As above, the dependent applications **12, 14, 16** (FIG. 1) described above may access each service offered by multi-modal content creation group **200** (e.g., transformation service **202**, import service **204**, and desktop integration service **210**) by making the appropriate request of and establishing a connection through the API assigned to that particular service. For this particular group, API **206** is assigned to the transformation service **202**, API **208** is assigned to the import service **204**, and API **212** is assigned to the desktop integration service **210**.

Referring to FIG. 5, the multi-channel deployment group **250**, which publishes files to one or more publishing channels, may include a static publishing service **252**, and a dynamic publishing service **254**.

The static publishing service **252** allows for proactive publishing of files based on predefined templates. Therefore, the structure and format of the file published (and the document produced) is defined ahead of time and is not varied depending on the content of the document. Additionally, the content itself is semi-dynamic, in that it changes periodically (e.g. a few times a week). An example of static documents generated using a static publishing service is a newsroom home page of a corporate web site in which one hundred press releases are currently being displayed. The home page is a collection of one hundred summary links, and each link leads to a press release detail page. A corporate communications officer can publish the home page and the one hundred detail pages by invoking the static publishing service, which merges the appropriate press release content with the detail page template to generate HTML. By generating the pages using static publishing, the communications officer ensures that web site visitors have fast page retrieval, since the content is already in HTML format and does not need to be regenerated for every website visitor.

The dynamic publishing service **254** allows for reactive publishing of files that are dynamically altered based on current conditions, user preferences, and query results, for example. In an online auction house that has one-hundred items for sale, a dynamic document may be created in response to a user query. For example, while one-hundred items may be offered, the user (an avid World War II buff) may only be interested in those items that relate to World War II. Therefore, the user would enter their search criteria and a dynamic document would be generated that includes fourteen items, each of which is related to World War II. This dynamically generated list, which itemizes the fourteen items, may also specify the starting bid, the current bid, and the auction ending time for each item. By generating this document with dynamic publishing services **254**, documents can be generated that more accurately reflect current conditions.

Regardless of whether the static publishing service **252** or dynamic publishing service **254** produce the file, the file can be published over various channels, such as: a web site; an email broadcast; a wireless broadcast; a syndication stream; and a printed publication, for example. The file can also be published in various formats, such as HTML, XML, and PDF, for example.

For web site publishing, the file being published may be posted to a website so that the file is accessible by various users and guests. If security or access is a concern, the file may be published on an intranet (which is not remotely accessible), or within a restricted access user section of a website.

For email broadcasts, the file can be published as an attachment to an email that is sent out to a distribution list of individual users. Alternatively, the file may be converted into a format (e.g., ASCII text, and HTML) that is easily incorporated into the body of an email.

For wireless broadcasts, the file can be transmitted to users over a wireless network. This file may be text-based (such as an email attachment sent to a wireless email device) or multimedia-based (such as a sound file sent to a cellular telephone).

For syndication streams, the file may be published on data streams that are text-based (such as streaming messages), audio-based (such as streaming audio), video-based (such as streaming video), or multimedia-based (such as streaming audio/video), for example.

For printed publications, the file being published may be printed on traditional printing systems (laser printers) and

distributed using traditional distribution paths (e.g., interoffice mail, courier, or the postal service, for example).

As above, the dependent applications **12**, **14**, **16** (FIG. 1) described above access each service offered by multi-channel deployment group **250** (e.g., static publishing service **252**, and dynamic publishing service **254**) by making the appropriate request of and establishing a connection through the API assigned to that particular service. For this particular group, API **256** is assigned to the static publishing service **252**, and API **258** is assigned to the dynamic publishing service **254**.

Referring to FIG. 6, a distributed computing system **300** is shown which incorporates the adaptive content platform described above. Distributed computing system **300** includes a local (or first) computing device **302** that executes one or more content-enabled dependent applications **12**, **14**. As described above, each of the dependent applications includes business logic and a user interface (FIG. 1).

A storage device **304** stores the individual instruction sets and subroutines of dependent applications **12**, **14**. Storage devices **304** may be a hard disk drive, a tape drive, an optical drive, a RAID array, a random access memory (RAM), or a read-only memory (ROM), for example. Local computing device **302** includes at least one processing unit (not shown) and main memory system (not shown).

A remote (or second) computing device **306** (e.g., a web server) executes the services layer **24**, as described above. Typically, services layer **24** is interfaced with a software infrastructure (not shown), which is interfaced with a data layer (not shown), which is interfaced with an OS layer (not shown).

A storage device **308** stores the individual instruction sets and subroutines of services layer **24** (and any additional required layers or infrastructure). Storage device **308** may be a hard disk drive, a tape drive, an optical drive, a RAID array, a random access memory (RAM), or a read-only memory (ROM), for example. Remote computing device **306** includes at least one processing unit (not shown) and main memory system (not shown).

Local computing device **302** and remote computing device **306** are interconnected with a network **310**, such as a LAN (local area network), WAN (wide area network), the Internet, or an intranet, for example.

While the above-described embodiment discusses the deployment of the services layer on a single second computer, other configurations are possible, such as those in which each service, or a group of services, is deployed on its own dedicated computer.

While the above-described embodiment describes a local and a remote computing device, this is not intended to define the physical location of either computing device and is merely intended to indicate that the second computing device is remote (i.e., separate) from the first computing device.

While the above-described embodiment discusses the use of content-enabled dependent applications, other configurations are possible, such as data-enabled dependent applications (i.e., those designed to manage data, as opposed to content).

While the above-described embodiment specifies a software infrastructure **24** that is an application server, other configurations are possible, such as a general-purpose operating system (e.g. UNIX, Windows 2000) or a special-purpose operating system (e.g. embedded OS, real time OS).

While the above-described embodiment illustrates the availability of three services and three dependent applica-

tions, the actual number of services and dependent applications can be adjusted based on system requirements.

Referring to FIG. 7, a distributed computing method **350** is shown. One or more content-enabled, dependent applications are executed **352** on a local computing device. Each dependent application includes a user interface and business logic. A services layer, which is interfaced with the dependent applications and a software infrastructure, is executed **354** on a remote computing device. The services layer provides one or more services that are usable by the content-enabled, dependent applications.

A data layer, which includes one or more databases, is interfaced **356** with the software infrastructure. A unique application programming interface is assigned **358** to each of the services. Dependent applications using a specific service make requests through the application programming interface assigned to that service.

The first and second computing devices are interconnected **360** with a network.

Referring to FIG. 8, a multi-tier software development method **400** is shown. One or more content-enabled, dependent applications are provided **402**, each of which includes a user interface and business logic. A services layer is provided **404**, which is interfaced with the one or more content-enabled dependent applications and provides one or more services that are usable by the content-enabled, dependent applications. The services layer is interfaced **406** with a software infrastructure interfaced **408** with a data layer. The data layer includes one or more databases.

The embodiments described herein are not limited to the embodiments described above; it may find applicability in any computing or processing environment. The embodiments may be implemented in hardware, software, or a combination of the two. For example, the embodiments may be implemented using circuitry, such as one or more of programmable logic (e.g., an ASIC), logic gates, a processor, and a memory.

The embodiments may be implemented in computer programs executing on programmable computers that each includes a processor and a storage medium readable by the processor (including volatile and non-volatile memory and/or storage elements). Each such program may be implemented in a high-level procedural or object-oriented programming language to communicate with a computer system. However, the programs can be implemented in assembly or machine language. The language may be a compiled or an interpreted language.

Each computer program may be stored on an article of manufacture, such as a storage medium (e.g., CD-ROM, hard disk, or magnetic diskette) or device (e.g., computer peripheral), that is readable by a general or special purpose programmable computer for configuring and operating the computer when the storage medium or device is read by the computer to perform the functions of the embodiments. The embodiments may also be implemented as a machine-readable storage medium, configured with a computer program, where, upon execution, instructions in the computer program cause a machine to operate to perform the functions of the embodiments described above.

The embodiments described above may be used in a variety of applications. Although the embodiments are not limited in this respect, the embodiments may be implemented with memory devices in microcontrollers, general purpose microprocessors, digital signal processors (DSPs), reduced instruction-set computing (RISC), and complex instruction-set computing (CISC), among other electronic components.

9

The embodiments described above may also be implemented using integrated circuit blocks referred to as main memory, cache memory, or other types of memory that store electronic instructions to be executed by a microprocessor or store data that may be used in arithmetic operations.

A number of embodiments of the invention have been described. Nevertheless, it will be understood that various modifications may be made without departing from the spirit and scope of the embodiments described above.

What is claimed is:

1. An adaptive content platform that is stored on a storage medium having a multi-tiered software architecture, comprising:

a services layer that includes at least repository services, workflow services, file handling services, and publishing services;

a data layer for providing data services to the adaptive content platform with such data services including at least access to database storage and file storage that contain content;

an application server layer that connects to a dependent applications layer through the services layer and retrieves content from the data layer, with the application server layer for handling operations of the dependent applications within the adaptive content platform;

the dependent applications layer that includes a plurality of content-enabled dependent applications that connect to the services layer and a web layer, and whose operations are controlled by the application server layer, with each content-enabled dependent application being capable of being interfaced with services of the services layer such that specific content from the services layer is combined to define and build a combination application that emulates a standalone independent application; and

the web layer that connects to the dependent applications layer for enabling users to interface with the adaptive content platform.

2. The adaptive content platform of claim 1, wherein the services layer includes a unique application programming interface for each of the services, wherein the content-enabled dependent applications using a specific service may make requests through the unique application programming interface assigned to that service.

3. The adaptive content platform of claim 1, wherein the repository services include being part of content management services for storing and managing files, with the content management services further including search services.

4. The adaptive content platform of claim 3, wherein the repository service includes being used for storing files.

5. The adaptive content platform of claim 4, wherein the search service includes being used for allowing users to search files stored by the repository services.

6. The adaptive content platform of claim 1, wherein the workflow services include being part of workflow and collaboration services for managing projects and users, with the workflow and collaboration services further including collaboration services.

7. The adaptive content platform of claim 6, wherein the workflow and collaboration services include workflow services for managing the workflow of files.

8. The adaptive content platform of claim 6, wherein the workflow and collaboration services include user management and authentication services for managing users and user groups.

10

9. The adaptive content platform of claim 6, wherein the workflow and collaboration services include events and notification services for managing and broadcasting notifications to users that were generated by one or more services.

10. The adaptive content platform of claim 1, wherein the file handling services include multi-modal content creation services for manual creation and automated importation and conversion of files.

11. The adaptive content platform of claim 10, wherein the multi-modal content creation services include transformation and content services for converting the formats of files.

12. The adaptive content platform of claim 10, wherein the multi-modal content creation services include desktop integration services for manual contribution of content.

13. The adaptive content platform of claim 10, wherein the multi-modal content creation services include-import services for facilitating file importation from external systems.

14. The adaptive content platform of claim 1, wherein publishing services include multi-channel deployment services for publishing files.

15. The adaptive content platform of claim 14, wherein the multi-channel deployment services include publishing services for publishing files to one or more publishing channels.

16. The adaptive content platform of claim 15, wherein the one or more publishing channels includes a web site.

17. The adaptive content platform of claim 15, wherein the one or more publishing channels includes an email broadcast.

18. The adaptive content platform of claim 15, wherein the one or more publishing channels includes a wireless broadcast.

19. The adaptive content platform of claim 15, wherein the one or more publishing channels includes a syndication stream.

20. The adaptive content platform of claim 15, wherein the one or more publishing channels includes a printed publication.

21. The adaptive content platform of claim 14, wherein the multi-channel deployment services include dynamic publishing services.

22. The adaptive content platform of claim 14, wherein the multi-channel deployment services include static publishing services.

23. The adaptive content platform of claim 1, data layer that is interfaced with the application server, with the data layer including one or more databases.

24. A multi-tiered software architecture that is stored on a storage medium, comprising:

a services layer that includes at least repository services, with the repository services being capable of connection to database and file storage that contain content for the adaptive content platform;

a dependent applications layer that includes a plurality of content-enabled dependent applications that connect to the services layer, with each content-enabled dependent application being capable of being interfaced with the repository services of the services layer such that specific content from at least the repository services is combined to define and build a combination application that emulates a standalone independent application; and

a web layer that connects to the dependent applications layer for enabling users to interface with the adaptive content-platform.

11

25. The adaptive content platform of claim 24, wherein the services layer includes a plurality of services including at least repository services, workflow services, file handling services, and publishing services.

26. The adaptive content platform of claim 25, wherein the services layer includes a unique application programming interface for each of the services through which dependent applications using a specific service make requests.

27. The adaptive content platform of claim 25, wherein the repository services include being part of content management services for storing and managing files, with the content management services further including searching services.

28. The adaptive content platform of claim 25, wherein the store the services store the workflow and collaboration services for managing projects and users, with the workflow and collaboration services further including collaboration services.

29. The adaptive content platform of claim 25, wherein the file store the include multi-modal content creation services for automated importation and conversion of files.

30. The adaptive content platform of claim 25, wherein the publishing services include multi-channel deployment services for publishing files to one or more publishing channels.

31. The adaptive content platform of claim 30, wherein the one or more publishing channels includes a web site.

32. The adaptive content platform of claim 30, wherein the one or more publishing channels includes an email broadcast.

33. The adaptive content platform of claim 30, wherein the one or more publishing channels includes a wireless broadcast.

34. The adaptive content platform of claim 30, wherein the one or more publishing channels includes a syndication stream.

35. The adaptive content platform of claim 30, wherein the one or more publishing channels includes a printed publication.

36. The adaptive content platform of claim 24, further comprising an application server layer that connects to the dependent applications layer through the services layer and retrieves content from a data layer that includes the database and file storage, with the applications server layer for handling operations of the content-enabled dependent applications within the content platform; and

the data layer that is interfaced with the software infrastructure and application server layer, with the data layer including one or more databases.

37. An adaptive content platform that is stored on a storage medium having a multi-tiered software architecture, comprising:

an application server that connects to a plurality of content-enabled dependent applications through a services layer and retrieving content from a data layer, with the applications server for handling operations of the content-enabled dependent applications within the content platform;

the plurality of dependent applications, with each content-enabled dependent application being capable of being interfaced with services of the services layer such that specific content from the services layer is combined to define and build a combination application that emulates a standalone independent application;

the services layer that includes at least repository services, with the services layer connecting to a plurality of

12

content-enabled dependent applications and the application server, and with the repository services being capable of interfacing with each of the content-enabled dependent applications such that specific content from at least the repository services is combined to define and build the combination application that emulates a standalone independent application.

38. The adaptive content platform of claim 37, wherein the services layer includes a plurality of services including at least repository services, workflow services, file handling services, and publishing services, with each service including a unique application programming interface through which content-enabled dependent applications using a specific service make requests.

39. The adaptive content platform of claim 38, further comprising a data layer that connects to the application server, with the data layer including one or more databases.

40. A distributed computing system, comprising:

a first computing device for executing

a plurality of content-enabled dependent applications, with each content-enabled dependent application being capable of being interfaced with services of a services layer; and

a second computing device connecting to the first computing device for executing

a services layer that includes at least repository services, with the services layer connecting to the content-enabled dependent applications and an application server, and with the repository services being capable of interfacing with each of the content-enabled dependent applications such that specific content from at least the repository services is combined to define and build a combination application that emulates a standalone independent application, the application server for handling operations of the content-enabled dependent applications within the distributed computing system and retrieving content from a data layer, and

the data layer for containing content that connects to the application server.

41. The distributed computing system of claim 40, wherein the services layer includes a plurality of services including at least repository services, workflow services, file handling services, and publishing services.

42. The distributed computing system of claim 41, wherein the data layer including one or more databases.

43. The distributed computing system of claim 41, wherein the services layer includes a unique application programming interface for each of the services through which content-enabled dependent applications using a specific service make requests.

44. The distributed computing system of claim 40, further comprising a network for interconnecting the first and second computing devices.

45. The distributed computing system of claim 44, wherein the network is a local area network.

46. The distributed computing system of claim 44, wherein the network is the Internet.

47. The distributed computing system of claim 44, wherein the network is an intranet.

48. The distributed computing system of claim 40, wherein the second computing device is a web server.

13

49. A distributed computing system, comprising:
 a local computing device for executing
 a plurality of content-enabled dependent applications,
 with each content-enabled dependent application
 being capable of being interfaced with services of a
 services layer; and
 a remote computing device connecting to the local com-
 puting device for executing
 the services layer that includes at least repository
 services, with the services layer connecting to the
 content-enabled dependent applications and an
 application server, and with the repository services
 being capable of interfacing with each of the content-
 enabled dependent applications such that specific
 content from at least the repository services is com-
 bined to define and build a combination application
 that emulates a standalone independent application,
 the application server for handling operations of the
 content-enabled dependent applications with the dis-
 tributed computing system and retrieving content
 from a data layer, and
 the data layer for containing content that connects to
 the application server.

50. The distributed computing system of claim 49,
 wherein the services layer includes a plurality of services
 including at least repository services, workflow services, file
 handling services, and publishing services.

51. The distributed computing system of claim 50,
 wherein the data layer including one or more databases.

52. The distributed computing system of claim 50,
 wherein the services layer includes a unique application
 programming interface for each of the services through
 which content-enabled dependent applications using a spe-
 cific service make requests.

53. The distributed computing system of claim 49, further
 comprising a network for interconnecting the local and
 remote computing devices.

54. The distributed computing system of claim 53,
 wherein the network is a local area network.

55. The distributed computing system of claim 53,
 wherein the network is the Internet.

56. The distributed computing system of claim 53 to
 wherein the network is an intranet.

57. The distributed computing system of claim 53,
 wherein the remote computing device is a web server.

58. A distributed computing method, comprising the steps
 of:
 executing, on a local computing device, a plurality of
 content-enabled; dependent applications, with each
 content-enabled dependent application being capable
 of being interfaced with services of a services layer;
 and
 executing, on a remote computing device,
 a services layer that includes at least repository ser-
 vices, with the services layer connecting to the
 content-enabled dependent applications and an
 application server, with the repository services being
 capable of interfacing with each of the content-
 enabled dependent applications such that specific
 content from at least repository services is combined
 to define and build a combination application so each
 content-enabled dependent application will execute
 as a standalone independent application,
 an application server for handling operations of the
 content-enabled dependent applications within the
 distributed computing system and retrieving content
 from a data layer, and

14

the data layer for containing content that connects to
 the application server.

59. The distributed computing method of claim 58,
 wherein the services layer includes a plurality of services
 including at least repository services, workflow services, file
 handling services, and publishing services.

60. The distributed computing method of claim 59,
 wherein the data layer including one or more databases.

61. The distributed computing method of claim 59, further
 comprising assigning a unique application programming
 interface to each of the services through which each content-
 enabled dependent applications using a specific service
 make requests.

62. The distributed computing method of claim 58, further
 comprising interconnecting the first and second computing
 devices with a network.

63. The distributed computing method of claim 62,
 wherein the network is a local area network.

64. The distributed computing method of claim 62,
 wherein the network is the Internet.

65. The distributed computing method of claim 62,
 wherein the network is an intranet.

66. The distributed computing method of claim 62,
 wherein the remote computing device is a web server.

67. A multi-tier software deployment method, comprising
 the steps of:
 providing a services layer that includes at least repository
 services, with the repository services being capable of
 connection to database and file storage containing
 content;
 providing a plurality of content-enabled dependent appli-
 cations that connect to the services layer, with each
 content-enabled dependent application being interfaced
 with the repository services of the services layer such
 that specific content from at least the repository ser-
 vices is combined to define and build a combination
 application so that the content-enabled dependent
 application will execute as a standalone independent
 application.

68. The multi-tier software deployment method of claim
 67, further comprising interfacing the services layer with an
 application server, with the application server for handling
 operations of the content-enabled dependent applications
 within the distributed computing system and retrieving
 content from a data layer, and the data layer containing
 content that connects to the application server.

69. The multi-tier software deployment method of claim
 68, wherein the data layer including one or more databases.

70. A computer program product residing on a computer
 readable medium having a plurality of instructions stored
 thereon which, when executed by a processor, cause that
 processor to:
 execute, on a local computing device,
 a plurality of content-enabled, dependent applications,
 with each content-enabled dependent application
 being capable of being interfaced with services of a
 services layer; and
 execute, on a remote computing device,
 a services layer that includes at least repository ser-
 vices, with the services layer connecting to the
 content-enabled dependent applications and an
 application server, with the repository services being
 capable of interfacing with each of the content-
 enabled dependent applications such that specific
 content from at least the repository services is com-
 bined to define and build a combination application

15

so each content-enabled dependent application will execute as a standalone independent application, the application server for handling operations of the content-enabled dependent applications within the distributed computing system and retrieving content from a data layer, and the data layer for containing content that connects to the application server.

71. The computer program product of claim 70, wherein the services layer includes a plurality of services including at least repository services, workflow services, file handling services, and publishing services.

72. The computer program product of claim 71, wherein the data layer including one or more databases.

73. The computer program product of claim 70, further comprising instructions to assign a unique application programming interface to each of the services through which content-enabled dependent applications using a specific service make requests.

74. A computer-based, enterprise level system that includes an adaptive content platform having a multi-tiered software architecture, comprising:

a services layer for providing enterprise-wide services that includes at least repository services the, workflow services, file handling services, and publishing services;

an enterprise-wide data layer for providing data services to the adaptive content platform with such data services including at least access to database storage and file storage that contain content;

an enterprise-wide application server layer that connects to a dependent applications layer through the services layer and retrieves content from the data layer, with the application server layer for handling operations of the dependent applications within the adaptive content platform;

the dependent applications layer that includes a plurality of content-enabled dependent applications that connect to the services layer and a web layer, and whose operations are controlled by the application server layer, with each content-enabled dependent application being capable of being interfaced with enterprise-wide services of the services layer such that specific content from the enterprise-wide services is combined to define and build a combination application that emulates a standalone independent application; and the web layer that connects to the dependent applications layer for enabling users to interface with the adaptive content platform.

75. The computer-based, enterprise level system of claim 74, wherein the enterprise-wide services layer includes a unique application programming interface for each of the services, wherein the content-enabled dependent applications using a specific service may make requests through the unique application programming interface assigned to that service.

76. The computer-based, enterprise level system of claim 74, wherein the repository services include being part of content management services for storing and managing files, with the content management services further including search services.

77. The computer-based, enterprise level system of claim 76, wherein the repository service includes being used for storing files.

78. The computer-based, enterprise level system of claim 77, wherein search services include being used for allowing users to search files stored by the repository services.

16

79. The computer-based, enterprise level system of claim 74, wherein the workflow services include being part of workflow and collaboration services for managing projects and users, with the workflow and collaboration services further including collaboration services.

80. The computer-based, enterprise level system of claim 79, wherein the workflow and collaboration services includes workflow services for managing the workflow of files.

81. The computer-based, enterprise level system of claim 79, wherein the workflow and collaboration services includes user management and authentication services for managing users and user groups.

82. The computer-based, enterprise level system of claim 79, wherein the workflow and collaboration services include events and notification service for managing and broadcasting notifications to users that were generated by one or more services.

83. The computer-based, enterprise level system of claim 74, wherein the file handling services includes multi-modal content creation services for manual creation and automated importation and conversion of files.

84. The computer-based, enterprise level system of claim 83, wherein the multi-modal content creation services include transformation and content services for converting the formats of files.

85. The computer-based, enterprise level system of claim 83, wherein the multi-modal content creation services include desktop integration services for manual contribution of content.

86. The computer-based, enterprise level system of claim 83, wherein the multi-modal content creation services include import services for facilitating file importation from external systems.

87. The computer-based, enterprise level system of claim 74, wherein the publishing services include multi-channel deployment services for publishing files.

88. The computer-based, enterprise level system of claim 87, wherein the multi-channel deployment services include publishing services for publishing files to one or more publishing channels.

89. The computer-based, enterprise level system of claim 88, wherein the one or more publishing channels includes a web site.

90. The computer-based, enterprise level system of claim 88, wherein the one or more publishing channels includes an email broadcast.

91. The computer-based, enterprise level system of claim 88, wherein the one or more publishing channels includes a wireless broadcast.

92. The computer-based, enterprise level system of claim 88, wherein the one or more publishing channels includes a syndication stream.

93. The computer-based, enterprise level system of claim 88, wherein the one or more publishing channels includes a printed publication.

94. The computer-based, enterprise level system of claim 87, wherein the multi-channel deployment services include dynamic publishing services.

95. The computer-based, enterprise level system of claim 87, wherein the multi-channel deployment services include static publishing services.

96. The computer-based, enterprise level system of claim 74, data layer that is interfaced with the application server, with the data layer including one or more databases.

17

97. A computer-based, enterprise level system that includes an adaptive content platform having a multi-tiered software architecture, comprising:

- a services layer for providing enterprise-wide services that includes at least repository services, with the repository services being capable of connection to database and file storage that contain content for the adaptive content platform;
- a dependent applications layer that includes a plurality of content-enabled dependent applications that connect to the services layer, with each content-enabled dependent application being capable of being interfaced with the repository services of the services layer such that specific content from at least the repository services is combined to define and build a combination application that emulates a standalone independent application; and
- a web layer that connects to the dependent applications layer for enabling users to interface with the adaptive content platform.

98. The computer-based, enterprise level system of claim 97, wherein the services layer includes a plurality of services including at least repository services, workflow services, file handling services, and publishing services.

99. The computer-based, enterprise level system of claim 98, wherein the services layer includes a unique application programming interface for each of the services through which dependent applications using a specific service make requests.

100. The computer-based, enterprise level system of claim 98, wherein the repository services include being part of content management services for storing and managing files, with the content management services further including searching services.

101. The computer-based, enterprise level system of claim 98, wherein the workflow services include being part of workflow and collaboration services for managing projects and users, with the workflow and collaboration services further including collaboration services.

102. The computer-based, enterprise level system of claim 98, wherein the file handling services include multi-modal content creation services for automated importation and conversion of files.

103. The computer-based, enterprise level system of claim 98, wherein the publishing services include multi-channel deployment services for publishing files to one or more publishing channels.

104. The computer-based, enterprise level system of claim 103, wherein the one or more publishing channels includes a web site.

105. The computer-based, enterprise level system of claim 103, wherein the one or more publishing channels includes an email broadcast.

106. The computer-based, enterprise level system of claim 103, wherein the one or more publishing channels includes a wireless broadcast.

107. The computer-based, enterprise level system of claim 103, wherein the one or more publishing channels includes a syndication stream.

108. The computer-based, enterprise level system of claim 103, wherein the one or more publishing channels includes a printed publication.

109. The computer-based, enterprise level system of claim 97, further comprising

- an enterprise-wide application server layer that connects to the dependent applications layer through the services layer and retrieves content from an enterprise-wide data

18

layer that include the database and file storage, with the applications server layer for handling operations of the content-enabled dependent applications within the content platform; and

- the enterprise-wide data layer that is interfaced with the software infrastructure and application server layer, with the data layer including one or more databases.

110. A computer-based, enterprise level system that includes an adaptive content platform having a multi-tiered software architecture, comprising:

- an enterprise-wide application server that connects to a plurality of content-enabled dependent applications through a services layer and retrieve content from a data layer, with the applications server for handling operations of the content-enabled dependent applications within the content platform;

the plurality of dependent applications, with each content-enabled dependent application being capable of being interfaced with services of the services layer such that specific content from the services layer is combined to define and build a combination application that emulates a standalone independent application; and

the services layer for providing enterprise-wide services that includes at least repository services, with the services layer connecting to a plurality of content-enabled dependent applications and the application server, and with the repository services being capable of interfacing with the each of the dependent applications such that specific content from at least the repository services is combined to define and build the combination application that emulates a standalone independent application.

111. The computer-based, enterprise level system of claim 110, wherein the services layer includes a plurality of services including at least repository services, workflow services, file handling services, and publishing services, with each service including a unique application programming interface through which content-enabled dependent applications using a specific service make.

112. The computer-based, enterprise level system of claim 111, further comprising an enterprise-wide data layer that connects to the application server, with the data layer including one or more databases.

113. An enterprise-wide, distributed computing system, comprising:

- a first computing device for executing
 - a plurality of content-enabled dependent applications, with each content-enabled dependent application being capable of being interfaced with services of a services layer; and
- a second computing device connecting to the first computing device for executing
 - a services layer for providing enterprise-wide services that includes at least repository services, with the services layer connecting to the content-enabled dependent applications and an enterprise-wide application server, and with the repository services being capable of interfacing with each of the content-enabled dependent applications such that specific content for at least the repository services is combined to define and build a combination application that emulates a standalone independent application,

19

the application server for handling operations of the content-enabled dependent applications within the distributed computing system and retrieving content for a data layer, and

the data layer containing content that connects to the application server.

114. The enterprise-wide, distributed computing system of claim 113, wherein the services layer includes a plurality of services including at least repository services, workflow services, file handling services, and publishing services.

115. The enterprise-wide, distributed computing system of claim 114, where the data layer including one or more databases.

116. The enterprise-wide, distributed computing system of claim 114, wherein the services layer includes a unique application programming interface for each of the services through which content-enabled dependent applications using a specific service make requests.

117. The enterprise-wide, distributed computing system of claim 113, further comprising a network for interconnecting the first and second computing devices.

118. The enterprise-wide, distributed computing system of claim 117, wherein the network is a local area network.

119. The enterprise-wide, distributed computing system of claim 117, wherein the network is the Internet.

120. The enterprise-wide, distributed computing system of claim 117, wherein the network is an intranet.

121. The enterprise-wide, distributed computing system of claim 113, wherein the second computing device is a web server.

122. An enterprise-wide, distributed computing system, comprising:

a local computing device for executing

a plurality of content-enabled dependent applications, with each content-enabled dependent application being capable of interfaced with services of a services layer; and

a remote computing device connecting to the local computing device for executing

the services layer for providing enterprise-wide services that includes at least repository services, with the services layer connecting to the content-enabled dependent applications and an enterprise-wide application server, and with the repository services being capable of interfacing with each of the content-enabled dependent applications such that specific content from at least the repository services is combined to define and build a combination application that emulates a standalone independent application, the application server for handling operations of the content-enabled dependent applications with the distributed computing system and retrieving content from a data layer, and

the data layer for containing content that connects to the application server.

123. The enterprise-wide, distributed computing system of claim 122, wherein the services layer includes a plurality of services including at least repository services, workflow services, file handling services, and publishing services.

124. The enterprise-wide, distributed computing system of claim 123, wherein the data layer including one or more databases.

125. The enterprise-wide, distributed computing system of claim 123, wherein the services layer includes a unique application programming interface for each of the services through which content-enabled dependent applications using a specific service make requests.

20

126. The enterprise-wide, distributed computing system of claim 122, further comprising a network for interconnecting the local and remote computing devices.

127. The enterprise-wide, distributed computing system of claim 126, wherein the network is a local area network.

128. The enterprise-wide, distributed computing system of claim 126, wherein the network is the Internet.

129. The enterprise-wide, distributed computing system of claim 126, wherein the network is an intranet.

130. The enterprise-wide, distributed computing system of claim 126, wherein the remote computing device is a web server.

131. An enterprise-wide, distributed computing method, comprising the steps of:

executing, on a local computing device, a plurality of content-enabled, dependent applications, with each content-enabled dependent application being capable of being interfaced with services of a services layer; and

executing, on a remote computing device,

a services layer for providing enterprise-wide services that includes at least repository services, with the services layer connecting to the content-enabled dependent applications and an application server, with the repository services being capable of interfacing with each of the content-enabled dependent applications such that specific content from at least the repository services is combined to define and build a combination application so each content-enabled dependent application will execute as a standalone independent application,

an enterprise-wide application server for handling operations of the content-enabled dependent applications within the distributed computing system and retrieving content from a data layer, and

the data layer containing content that connects to the application server.

132. The enterprise-wide, distributed computing method of claim 131, wherein the services layer includes a plurality of services including at least repository services, workflow services, file handling services, and publishing services.

133. The enterprise-wide, distributed computing method of claim 132, wherein the data layer including one or more databases.

134. The enterprise-wide, distributed computing method of claim 132, further comprising assigning a unique application programming interface to each of the services through which each content-enabled dependent applications using a specific service make requests.

135. The enterprise-wide, distributed computing method of claim 131, further comprising interconnecting the first and second computing devices with a network.

136. The enterprise-wide, distributed computing method of claim 135, wherein the network is a local area network.

137. The enterprise-wide, distributed computing method of claim 135, wherein the network is the Internet.

138. The enterprise-wide, distributed computing method of claim 135, wherein the network is an intranet.

139. The enterprise-wide, distributed computing method of claim 135, wherein the remote computing device is a web server.

140. A deployment method for a computer-based, enterprise level system that includes multi-tier software, comprising the steps of:

providing a services layer for providing enterprise-wide services that includes at least repository services, with

21

the repository services being capable of connection to database and file storage containing content;
 provide a plurality of content-enabled, dependent applications that connect to the services layer, with each content-enabled dependent application being interfaced with the repository services of the services layer such that specific content from at least the repository services is combined to define and build a combination application so that the content-enabled dependent application will execute as a standalone independent.

141. The method of claim **140**, further comprising interfacing the services layer with an enterprise-wide application server, with the application server for handling operations of the content-enabled dependent applications within the distributed computing system and retrieving content for a data layer, and the data layer containing content that connects to the application server.

142. The method of claim **141**, wherein the data layer including one or more databases.

143. An enterprise-wide, computer-based system with a computer program product residing on a computer readable medium having a plurality of instructions stored thereon which, when executed by the processor, cause that processor to:

execute, on a local computing device,
 a plurality of content-enabled, dependent applications,
 with each content-enabled dependent application being capable of being interfaced with services layer;
 and
 execute, on a remote computing device,

22

an enterprise-wide services layer that includes at least repository services, with the services layer connecting to the content-enabled dependent applications and an enterprise-wide application server, with the repository services being capable of interfacing with each of the content-enabled dependent applications such that specific content from at least the repository services is combined to define and build a combination application so each content-enabled dependent application will execute as a standalone independent application,

the application server for handling operations of the content-enabled dependent applications within the distributed computing system and retrieving content for a data layer, and

the data layer for containing content that connects to the application server.

144. The system of claim **143**, wherein the services layer includes a plurality of services including at least repository services, workflow services, file handling services, and publishing services.

145. The system of claim **14**, wherein the data layer including one or more databases.

146. The system of claim **143**, further comprising instructions to assign a unique application programming interface to each of the services through which content-enabled dependent applications using a specific service make requests.

* * * * *