



(51) International Patent Classification:
G06F 15/16 (2006.01) *G06F 3/00* (2006.01)
G06F 15/17 (2006.01)

(21) International Application Number:
PCT/US2008/080830

(22) International Filing Date:
22 October 2008 (22.10.2008)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/981,794 22 October 2007 (22.10.2007) US
12/240,836 29 September 2008 (29.09.2008) US

(71) Applicant (for all designated States except US): **XCERION AKTIEBOLAG** [SE/US]; 2003 Western Avenue, Suite 500, Seattle, WA 98121 (US).

(72) Inventor; and

(75) Inventor/Applicant (for US only): **ARTHURSSON, Daniel** [SE/SE]; 2003 Western Avenue, Suite 500, Seattle, WA 98121 (US).

(74) Agent: **FEEKES, Clint, J.**; Christensen O'Connor Johnson Kindness PLLC, 1420 5th Avenue, Suite 2800, Seattle, WA 98101-2347 (US).

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

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM,

[Continued on next page]

(54) Title: GESTURE-BASED COLLABORATION

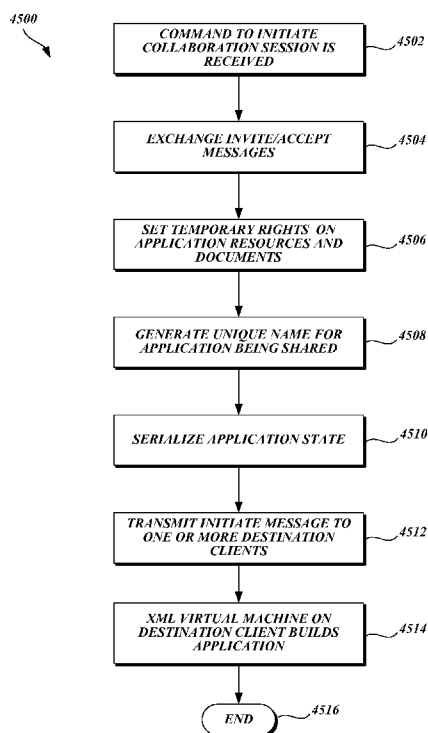


Fig.45.

(57) Abstract: Generally described, the present invention is directed to a network operating system that provides more effective ways of leveraging the connectivity of computer networks. In one embodiment, an XML virtual machine is implemented that accepts high-level application code written in an XML programming language as input. Functionality is provided to interpret or translate the application code written in an XML programming language into code that is suitable for execution across computer platforms. Moreover, the XML virtual machine supports the Model View Controller (MVC) design paradigm that facilitates true data abstraction from applications to a common data model. Multi-instance applications, with each instance potentially supporting multiple views, may be created and executed.



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, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

— *with amended claims (Art. 19(1))*

(88) Date of publication of the international search report:

16 July 2009

Date of publication of the amended claims:
2009

17 September

Published:

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

AMENDED CLAIMS
received by the International Bureau on 28 July 2009 (28.07.09)

The embodiments of the invention in which an exclusive property or privilege is claimed are defined as follows:

1. A method of actively sharing an application configured to be executed on both an originating and destination computing device, the method comprising:

initiating a collaboration session between the originating and destination computing devices, including:

exporting (4508-4512) the state of the executing application from the originating computing device to the destination computing device;

providing (4506) the destination computing device with rights to access the resources utilized by the application;

building (4514) the application on the destination computing device; and

in response to the occurrence of a state change in the application, propagating and interpreting (4600) a description of the state change on each computing device participating in the collaboration session.

2. The method as recited in Claim 1, wherein exporting (4508-4512) the state of the executing application from the originating computing device includes identifying the resources utilized by the application as specified in an application package.

3. The method as recited in Claim 1, wherein exporting (4508-4512) the state of the executing application includes delivering application resources to the destination computing device without requiring the application to have been previously installed on the destination computing device.

4. The method as recited in Claim 1, wherein the user drags-and-drops the visual rendering of an application dialog (4701) over the boundary of a visual display in order to initiate the collaboration session.

5. The method as recited in Claim 1, wherein an object associated with a user name (4708) displayed on a visual display is activated to initiate the collaboration session.

6. The method as recited in Claim 1, wherein exporting (4508-4512) the state of the executing application from the originating computing device to the destination computing device, includes:

maintaining an outbox special folder associated with the first user on a network file system;

maintaining an inbox special folder associated with the second user at the network file system;

wherein a server process is configured to monitor the outbox special folder associated with the first user, identify an incoming message to initiate a collaboration session located in the outbox special folder of the first user, and move the message to the inbox special folder of the second user.

7. The method as recited in Claim 1, wherein at least one resource in which the destination computing device is provided with access rights is identified dynamically from the process or view utilized by an application instance on the originating computing device.

8. The method as recited in Claim 1, wherein exporting (4508-4512) the state of the application, includes:

utilizing an identifier associated with the first user to generate (4508) a unique name for the collaboration session; and

wherein the resources of the application are associated with the unique name and temporary rights are provided to the second user to access application resources associated with the unique name.

9. The method as recited in Claim 1, wherein exporting (4508-4512) the state of the application includes serializing (4510) information from one or more objects associated with the application into a text or binary format.

10. The method as recited in Claim 9, wherein the information serialized (4510) from the one or more objects includes at least one variable describing a state of a user interface component selected from the group of a variable describing whether the user interface component is minimized, expanded, and selected.

11. The method as recited in Claim 1, wherein exporting (4508-4512) the state of the executing application includes configuring an object that models a user interface component to identify the data that the object is bound in an XBind.

12. The method as recited in Claim 1, wherein building (4514) the application on the destination computing device includes providing an XML virtual machine that accepts application code written in an XML programming language as input and causes the application code to be executed.

13. The method as recited in Claim 1, wherein building (4514) the application on the destination computing device includes utilizing a data type recognizer to identify the received document type as an application package.

14. A computer-readable medium bearing computer-executable instructions which, when executed by one or more computer processors, carries out a method of propagating a change affecting an application between computing devices, comprising:

establishing (4500) a collaboration session between a first and second computing device;

in response to the first computing device experiencing a change that affects the application:

generating (2006) a transaction that describes the change;

implementing (4604) the change locally on the first computing device;

causing (4606-4610) the transaction describing the change to be transmitted to the second computing device; and

interpreting (4614) the received transaction to implement the change on the second computing device.

15. The computer-readable medium as recited in Claim 14, wherein the type of collaboration session established causes the first and second computing devices to share a virtual desktop in which the position occupied by an application dialogue (4701) is shared across screen displays.

16. The computer-readable medium as recited in Claim 14, wherein the type of collaboration session established causes the presentation and position occupied by an application dialogue (4701) to be mirrored on both the first and second computing devices.

17. The computer-readable medium as recited in Claim 14, wherein the type of collaboration session established causes an application view to be cloned so that state changes and data updates affecting the application are propagated between the first computing devices, and wherein re-positioning of an application dialogue (4701) is not mirrored on both the first and second computing devices.

18. The computer-readable medium as recited in Claim 14, wherein establishing (4500) a collaboration session between a first and second computing device, includes:

identifying a dynamic resource used by an application instance executing on the first computing device; and

providing (4506) the second computing device with at least temporary rights to access the dynamic resource.

19. The computer-readable medium as recited in Claim 14, wherein implementing (4604) the state change locally on the first computing device includes providing a data update event bus that automatically propagates a data update from a document to a user interface component bound to the document.

20. The computer-readable medium as recited in Claim 14, wherein the state change is implemented (4604) locally on the first computing device asynchronously from the transaction being transmitted to the second computing device.

21. The computer-readable medium as recited in Claim 14, wherein interpreting (4614) the transaction on the second computing device includes deserializing XML data in the transaction and providing the deserialized data to an appropriate in-memory object.

22. A computing device (300) configured to participate in a collaboration session, comprising:

a processor (302);

a network interface (310) for sending data over the network while participating in the collaboration session;

a memory (304) having computer-readable media with computer-executable components for exchanging data in the collaboration session, including:

a data update bus (1254) configured to automatically propagate data updates from the applications' user interface to one or more documents bound to the application;

a state manager (4410) that automatically causes the state of the applications view to be described in one or more state documents;

a collaboration manager (4412) configured to:

identify the occurrence data updates affecting the one or more documents bound to the application and state changes affecting the visual display of the application; and

cause transactions describing the data updates and state change to be encapsulated and transmitted to a second computing device that is participating in the collaboration session.

23. The computing device (300) as recited in Claim 22, wherein the data update bus (1254) is further configured to cause the data update to be implemented locally by causing user interface components bound to a document to be notified of the data update.

24. The computing device as recited in Claim 22, wherein to identify the occurrence of state changes affecting the visual display of the application includes:

determining whether a user interface component that experienced a state change is a atomic component;

if the state change affects an atomic component, obtaining data describing the state change from an in-memory object; and

propagating data describing the state change to the one or more state documents.

25. The computing device as recited in Claim 22, wherein the memory (304) further includes a communication channel component for abstracting communications with a network service that manages the exchange of messages during the collaboration session.

26. The computing device as recited in Claim 22, wherein the collaboration manager (4410) is further configured to:

obtain an update message that describes a state change to the application that occurred on the second computing device; and

cause a user interface component to implement the state change that occurred on the second computing device.

27. Computer-readable media, computing devices, systems, methods, and apparatus as described and illustrated herein.