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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
21 July 2011 (21.07.2011)

PCT

(10) International Publication Number
WO 2011/086180 A1

(51) International Patent Classification:
G06F 9/445 (2006.01)

(21) International Application Number:
PCT/EP2011/050527

(22) International Filing Date:
17 January 2011 (17.01.2011)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10305054.8 18 January 2010 (18.01.2010) EP

(71) Applicant *(for all designated States except US)*: THOMSON LICENSING [FR/FR]; 1-5 rue Jeanne d'Arc, F-92130 Issy-les-Moulineaux (FR).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: MAETZ, Yves [FR/FR]; Thomson Multimedia R&d France, Technicolor - IP&L Dept., 1 Avenue de Belle Fontaine, F-35510 Cesson-Sévigné (FR). ELUARD, Marc [FR/FR]; Thomson Multimedia R&d France, Technicolor - IP&L Dept., 1 Avenue de Belle Fontaine, F-35510 Cesson-Sévigné (FR).

(74) Agent: RUELLAN-LEMONNIER, Brigitte; Technicolor, 1-5 rue Jeanne d'Arc, F-92443 ISSY LES MOULINEAUX cedex (FR).

(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, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, 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, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

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

[Continued on next page]

(54) Title: METHOD, SYSTEM AND DEVICE FOR EXECUTION OF A SOFTWARE APPLICATION

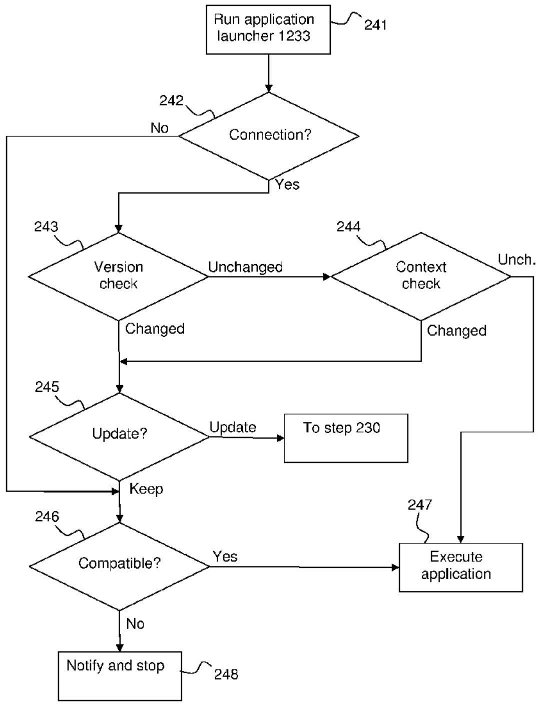

Figure 5

(57) Abstract: A method of executing a software application. A storage device (120) stores connection data (1211), an identifier (1212) of the software application, a context selector (1232), and an application launcher (1233). An execution device connects (231), using the connection data, to a server (130). The storage device transfers the context selector to the execution device, which executes (232) it to generate a first context for the execution device. The identifier and the first context are sent to the server from which the storage device receives the software application (1231), which is stored (234) with the first context. The application launcher is transferred to the execution device for execution (241). A second context of the execution device is generated and the storage device checks (244) if it matches the first context. If so, the storage device transfers the software application to the execution device where it is executed.

Declarations under Rule 4.17:

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

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

— *of inventorship (Rule 4.17(iv))*

Published:

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

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

METHOD, SYSTEM AND DEVICE FOR EXECUTION OF A SOFTWARE APPLICATION

TECHNICAL FIELD

The present invention relates generally to software, and in particular to software distribution.

BACKGROUND

This section is intended to introduce the reader to various aspects of art, which may be related to various aspects of the present invention that are described and/or claimed below. This discussion is believed to be helpful in providing the reader with background information to facilitate a better understanding of the various aspects of the present invention. Accordingly, it should be understood that these statements are to be read in this light, and not as admissions of prior art.

There are of course many existing ways of distributing software, usually based on having the user select the desired version (depending on the operating system) and then either downloading the software or installing it from a DVD, a USB key or the like. A first variant is to download an exact software image onto the target computer, but this causes problems if there is the slightest difference between the configuration of the target computer and the configuration for which the software image is intended. A second variant is to run an installation program that either requests the user to enter manually the configuration of the target computer or to have the installation program do this automatically. This second variant results in a software image that is an exact match for the target computer.

Various bells and whistles to the basic schemes are provided in for example US 2008/0046708, WO 2008/154426, US 2006/0218549, US 2006/075486 and US 2008/250476 without changing the essentials thereof.

A drawback with these solutions is that they are fairly inflexible. It can therefore be appreciated that there is a need for a more flexible solution for software installation. This invention provides such a solution.

SUMMARY OF INVENTION

In a first aspect, the invention is directed to a method of downloading a first software application in a system comprising a first device adapted to execute the first software application, the first device being connected to a second device that stores an identifier of the first software application. The second device obtains first configuration information for the first device; sends the identifier and the first configuration information to the first device for transmission to a server; receives the first software application from the server, the software application corresponding to the identifier and to the first configuration information; and transfers the software application to the first device.

In a first preferred embodiment, the second device verifies, prior to transferring the first software application, that second configuration information for the first device matches the first configuration information; the first software application is transferred only in case of positive verification. It is advantageous that the second configuration information is received from the first device.

In a second preferred embodiment, the first configuration information is obtained by transferring a configuration selection application to the first device; and receiving the first configuration information from the first device having executed the configuration selection application to generate the first configuration information for the first device.

In a third preferred embodiment, the second device stores the first configuration information and the software application.

In a fourth preferred embodiment, the second device sends to the first device connection data for connection to the server.

In a second aspect, the invention is directed to a first device in a system for executing a first software application. The system further comprises a second device adapted to execute the first software application. The first device is adapted to be connected to the second device. The first device comprises memory for storing an identifier of the first software application; and a processor adapted to obtain first configuration information for the second device; send the identifier and the first configuration information to the second device for transmission to a server; receive the first software application from the server, the first software application corresponding to the identifier and to the first configuration information; and transfer the first software application to the second device.

In a first preferred embodiment, the processor is further adapted to verify that second configuration information for the first device matches the first configuration information; and wherein the transfer is performed only in case of positive verification. It is advantageous that the processor is further adapted to receive the second configuration information from the second device.

In a second preferred embodiment, the processor is adapted to obtain the first configuration information by transferring a configuration selection application to the second device; and receiving the first configuration information from the second device having executed the configuration selection application to generate the first configuration information for the second device.

In a third preferred embodiment, the memory is further for storing the first configuration information and the software application.

In a fourth preferred embodiment, the processor is further adapted to send to the second device connection data for connection to the server.

BRIEF DESCRIPTION OF DRAWINGS

Preferred features of the present invention will now be described, by way of non-limiting example, with reference to the accompanying drawings, in which Figure 1 illustrates a system according to a preferred embodiment of the invention;

Figure 2 illustrates a flowchart describing the main steps of the software distribution method according to a preferred embodiment of the present invention; and

Figures 3-5 illustrate in greater detail main steps of the software distribution method according to a preferred embodiment of the present invention.

DESCRIPTION OF EMBODIMENTS

Figure 1 illustrates an exemplary system 100 implementing the invention. The system 100 comprises a target 110, a self-optimized application holder (hereinafter SAH) 120, and an application server 130.

The target 110 can be practically any kind of computer or device capable of performing calculations, such as a standard Personal Computer (PC). The target 110 comprises a processor 111, RAM memory 112, a first interface 113 for communication with the SAH 120, and a second interface 114 for communication with the application server 130. The communication with the application server 130 is advantageously performed via the Internet.

The SAH 120 comprises an interface 125 for releasable connection to and communication with the target 110, a processor 121 and non-volatile memory 123. The processor 121 is among other things adapted to store connection data 1211, an identifier for an application 1212 and a context of the target 110. The connection data 1211 comprises information necessary to establish a connection with the application server 130, such as the address of the server and, preferably, data permitting (mutual) authentication. This will be described in more detail hereinafter. A context is information concerning a hardware configuration, a software configuration or a hardware and software configuration. The non-volatile memory 123 is adapted to store application code 1231, a context selection application 1232 and an application launcher 1233, which also will be described further hereinafter.

In a preferred embodiment, the SAH 120 is implemented as a secure USB key wherein the non-volatile memory 123 is a flash memory, the processor 121 is a tamper-resistant processor, and the interface 125 is a USB interface (which naturally means that the first interface 113 of the target 110 also is a USB interface).

The application server 130 may be any kind of suitable server that is capable of communication with the target 110. The server 130 may host a single software application or a plurality of different software applications.

The present invention differs from the prior art in that a user does not buy an application directly. Instead, the user buys a unique SAH that does not store the code for the application, but generic applications and information that enables download from the application server, as will be seen hereinafter.

Figure 2 is a flowchart that describes the four main steps of the software distribution method according to a preferred embodiment of the present invention. Each of the four steps is described in more detail hereinafter. At first, the SAH for a specific software application is created, step 210. This is followed by transfer of the SAH, step 220, usually when the SAH is bought by a user. The software application is then downloaded, step 230, and executed, step 240.

Figure 3 illustrates the SAH creation step 210 of the method illustrated in Figure 2. The steps in Figure 3 are preferably performed during manufacture of the SAH. The connection data 1211 is stored in the processor 121, step 211. The communication data 1211 will be used for communication with the server 130. The application identifier 1212 is also stored in the processor 121, step 212. Then the context selection application 1232 and the application launcher 1233 are stored in the non-volatile memory 123, steps 213 and 214 respectively. It will be appreciated that these steps may be performed in practically any order.

During the SAH transfer step 220, the SAH 120 is provided to a user, usually in exchange for some kind of payment.

Figure 4 illustrates the application download step 230 of the method illustrated in Figure 2. This step may be performed in three cases: 1) the SAH 120 stores no application code 1231 (this is the case illustrated in Figure 2), 2) a new version of the application code – patches, upgrades, add-ons, etc. – is available, and 3) the context has changed, for example if the target has changed or if the SAH 120 has been connected to a different target.

In step 231, the SAH 120 uses the connection data 1211 to connect to the server 130, via the target 110, step 231. The SAH 120 provides the context selection application 1232 to the target 110 that launches this application. The context selection application 1232 then either automatically detects the context of the target 110 or requests the user to select the context, step 232. The context is the relevant hardware and software configuration of the target, such as for example characteristics of the operating system and the graphic card. This may for example be performed as described in US 2008/0046708. While automatic context detection is convenient, manual input by the user enables the user to select the context of a target that is different from the computer to which the SAH 120 is connected. The context 1213 is then sent to the SAH 120 that stores it in the processor 121 and to the server 130, which uses the received context to either select an already existing version of the software application, if such a version exists, or to build the version of the software application that is optimised for the target 110. This software application 1231 is then sent to the SAH 120 where it is stored in the non-volatile memory 123 and the context 1213 is stored in the processor 121, step 234. The SAH 120 also receives from the server 130 a version identifier for the software application. This version identifier may e.g., depending on the implementation, be stored as a subfield of the application identifier 1212 or the context 1213, or be stored separately, for instance in the header of the application code 1231.

In a preferred embodiment, to connect to the server 130, the SAH 120 also uses the data permitting authentication, which allows the server 130 to verify the authenticity of the SAH 120 and also allows integrity and confidentiality of transmitted data.

It is preferred that if an error is detected by the SAH, it enters an error state that lasts until the next connection to a target. This is to ensure the atomicity of the method. Naturally, other error management policies may also be used, but they are out of the scope of the present invention.

Figure 5 illustrates the application execution step 240 of the method illustrated in Figure 2. This step may be performed directly after completion of step 230 or when a SAH 120 that stores application code 1231 is connected to a computer.

First, the application launcher 1233 is provided to the target 110 where it is executed, step 241. A main task of the application launcher 1233 is to verify that the software application 1231 may be executed by the target 110. It is then verified, step 242, if the application launcher 1233 has a connection with the server 130. If this is not the case, then the method moves to compatibility verification step 246 described hereinafter. However, if there is a connection, the application launcher checks, step 243, using the application identifier 1212, if the version of the software application 1231 stored in the SAH 120 is the latest version, i.e. if the version provided by the server has changed when compared to the stored one. If the versions are the same, i.e. there is nothing new to download, then the method moves on to step 244 in which the context is checked. This step is also performed in case it is detected in step 242 that no connection exists.

The context check, step 244, verifies if the context of the target 110 (preferably calculated automatically as in step 232) matches the context 1213 stored in the processor 121 (i.e. if the contexts are identical or, depending on the implementation, if the context of the target is compatible with the stored context). It is for example possible that an application is particularly suited for a family of processors, in which case the application is compatible with these processors. If this is the case, i.e. if the contexts match, then the software application is uploaded from the non-volatile memory 123 and executed, step 247.

However, if there is no match, then the context has changed or is incompatible. This may be the case if the target 110 has changed hardware or software, or if the SAH 120 is connected to a different computer than the one for which the context was selected. It is preferred that it is indicated to the user that the context has changed and that an update may be in order. Upon reception of the input from the user, step 245, the method may move in two directions. If the user wants the update, then the method moves to step 230 and performs the necessary sub-steps. But if the user does not want to update the software application, or if such an update cannot be performed for example owing to the lack of a connection to the server, then it is verified, step 246, if the version of the software application is compatible with the context.

If the software version is compatible with the context, then the method executes the software application, step 247. On the other hand, if the software version is incompatible with the context, then this is indicated to the user and the method stops, step 248.

In a preferred embodiment, the SAH 120 implements the anti-theft mechanism described in WO 2008/129009. The anti-theft mechanism uses a RFID chip that can be in one of two states. In the first state, the RFID chip can interact with anti-theft portals and in addition the RFID chip refuses to provide information necessary for full use of the support to which it is attached. The RFID chip moves to the second state when the support is bought, i.e. when it is put in contact with a state-change device. In the second state, the RFID no longer interacts with anti-theft portals and it also provides the necessary information. An advantage is that the support does not work even if it stolen. As applied to the SAH, the RFID could for example securely store the connection data 1211 and the application identifier 1212; without these, the SAH will, at least in theory, not work.

It will be appreciated that the place of storage in the SAH, i.e. in the processor or in the non-volatile memory, is the preferred solution, but that this location may be altered depending on the implementation. It will also be appreciated that the present invention can provide a way of distributing software that is more flexible than prior art solutions.

Each feature disclosed in the description and (where appropriate) the claims and drawings may be provided independently or in any appropriate combination. Features described as being implemented in hardware may also be implemented in software, and vice versa. Reference numerals appearing in the claims are by way of illustration only and shall have no limiting effect on the scope of the claims.

CLAIMS

1. A method of downloading a first software application in a system comprising a first device (110) adapted to execute the first software application, the first device being connected to a second device (120), the second device (120) storing an identifier (1212) of the first software application, the method comprising the steps at the second device (120) of:

- obtaining first configuration information (1213) for the first device (110);
- sending the identifier (1212) and the first configuration information (1213) to the first device (110) for transmission to a server (130);
- receiving the first software application (1231) from the server (130), the software application (1231) corresponding to the identifier (1212) and to the first configuration information (1213); and
- transferring the software application (1231) to the first device (110).

2. The method of claim 1, further comprising the step, prior to the transferring step, of verifying (244) that second configuration information for the first device (110) matches the first configuration information (1213); and wherein the transferring step is performed only in case of positive verification.

3. The method of claim 2, wherein the second configuration information is received from the first device (110).

4. The method of claim 1, wherein the first configuration information (1213) is obtained by:

- transferring a configuration selection application (1232) to the first device (110); and
- receiving the first configuration information from the first device (110) having executed the configuration selection application (1232) to generate the first configuration information (1213) for the first device (110).

5. The method of claim 1, further comprising the step of storing (234) the first configuration information (1213) and the software application (1231).

6. The method of claim 1, further comprising the step of sending to the first device (110) connection data (1211) for connection to the server (130).

7. A first device (120) in a system for executing a first software application, the system further comprising a second device (110) adapted to execute the first software application, the first device (120) being connected to the second device (110), the first device (120) comprising:

- memory (121, 123) for storing an identifier (1212) of the first software application; and

- a processor (121) adapted to:

- obtain first configuration information (1213) for the second device (110);

- send the identifier (1212) and the first configuration information (1213) to the second device (110) for transmission to a server (130);

- receive the first software application (1231) from the server (130), the first software application (1231) corresponding to the identifier (1212) and to the first configuration information (1213); and

- transfer the first software application (1231) to the second device (110).

8. The first device of claim 7, wherein the processor is further adapted to verify that second configuration information for the first device (110) matches the first configuration information (1213); and wherein the transfer is performed only in case of positive verification.

9. The first device of claim 8, wherein the processor is further adapted to receive the second configuration information from the execution device (110).

10. The first device of claim 7, wherein the processor is adapted to obtain the first configuration information (1213) by:

- transferring a configuration selection application (1232) to the execution device (110); and

- receiving the first configuration information from the execution device (110) having executed the configuration selection application (1232) to generate the first configuration information (1213) for the execution device (110).

11. The first device of claim 7, wherein the memory (121, 123) is further for storing (234) the first configuration information (1213) and the software application (1231).

12. The first device of claim 7, wherein the processor (121) is further adapted to send to the execution device (110) connection data (1211) for connection to the server (130).

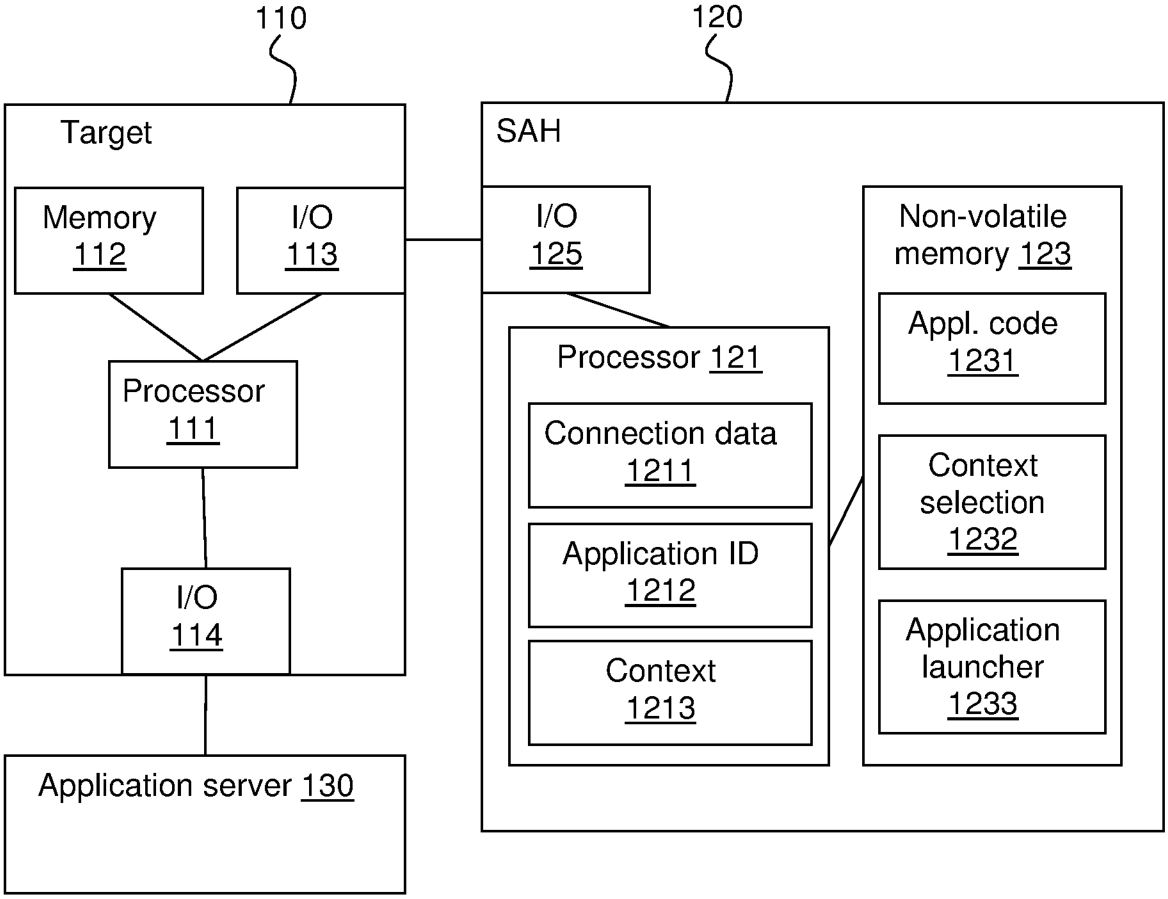

Figure 1

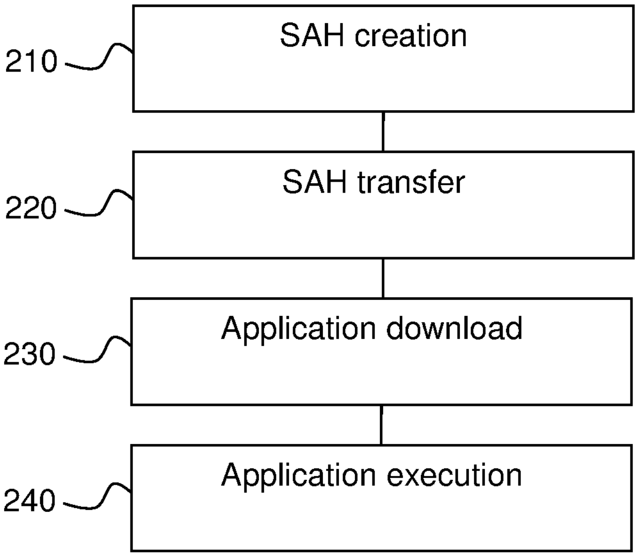

Figure 2

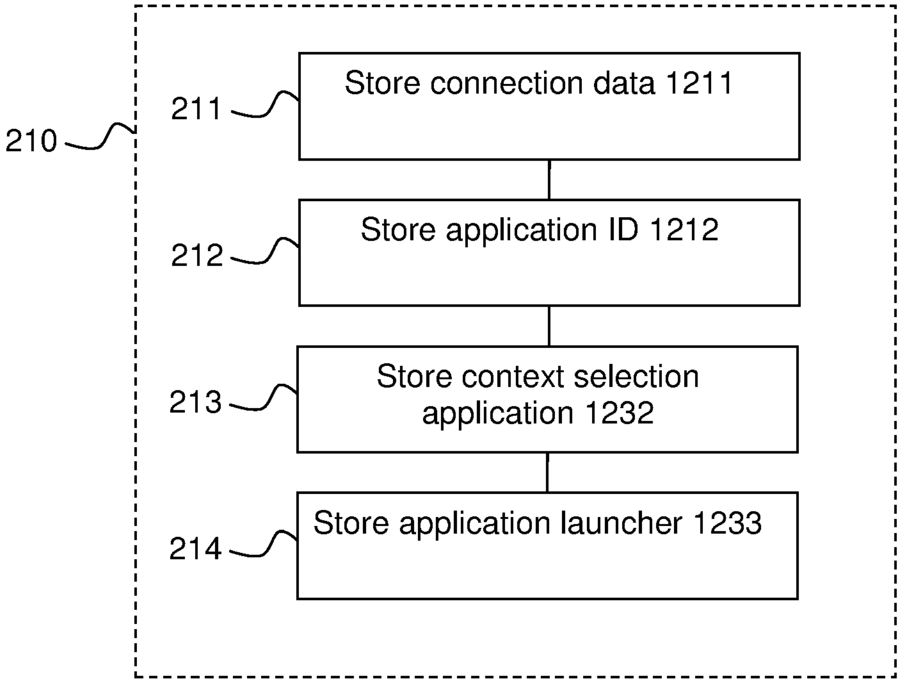
210
211
Store connection data 1211
212
Store application ID 1212
213
Store context selection application 1232
214
Store application launcher 1233

Figure 3

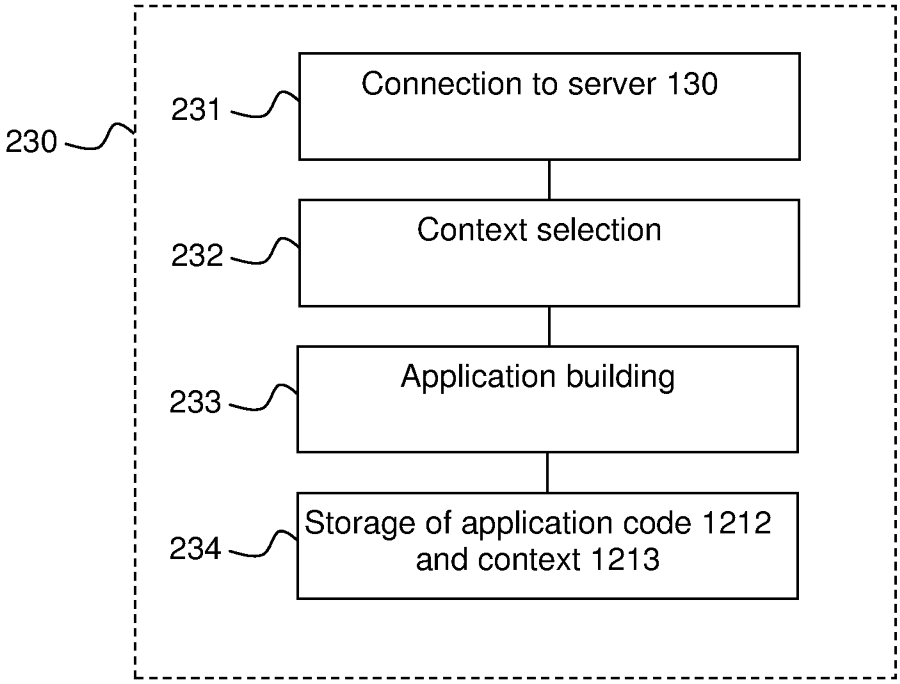

Figure 4

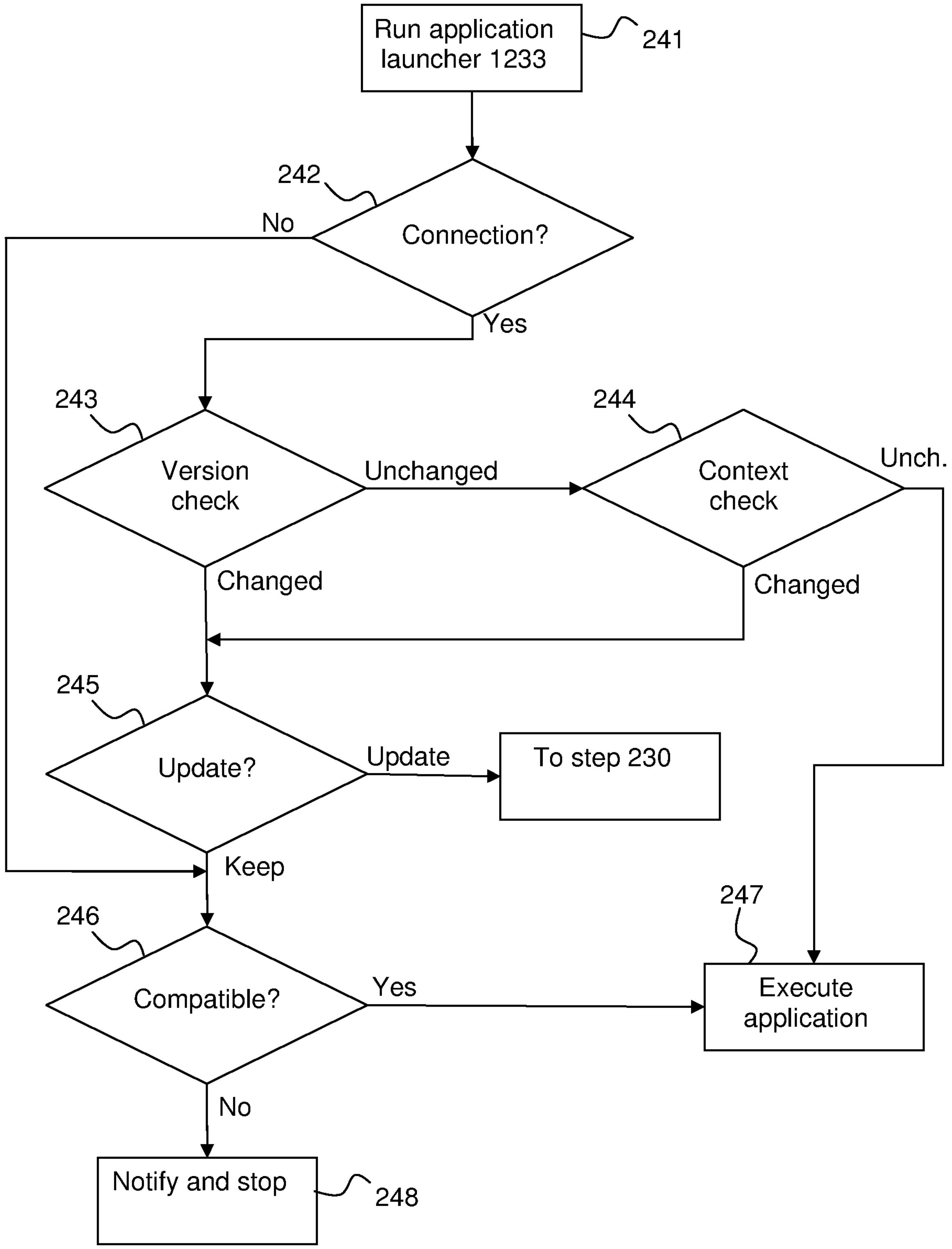
Run application launcher 1233
241
242
Connection?
No
Yes
243
Version check
Unchanged
Changed
244
Context check
Unch.
Changed
245
Update?
Update
To step 230
Keep
246
Compatible?
Yes
No
247
Execute application
Notify and stop
248

Figure 5

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2011/050527

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F9/445
ADD.

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 H04M H04L

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, WPI Data, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2007/118617 A1 (LEE JANGWON [US] ET AL) 24 May 2007 (2007-05-24) abstract paragraph [0007] paragraph [0010] - paragraph [0014] paragraph [0032] - paragraph [0034] paragraph [0036] - paragraph [0046] paragraph [0048] - paragraph [0049] paragraph [0052] - paragraph [0076] claims 1-57; figures 1-12 ----- -/--	1-12

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

☒ See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier 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 18 May 2011	Date of mailing of the international search report 26/05/2011
Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Authorized officer Eftimescu, Nicolae

1

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 1 832 976 A1 (FUJITSU LTD [JP]) 12 September 2007 (2007-09-12) abstract paragraph [0007] - paragraph [0011] paragraph [0013] - paragraph [0027] paragraph [0050] - paragraph [0101] paragraph [0103] - paragraph [0121] paragraph [0123] - paragraph [0158] paragraph [0161] - paragraph [0166] claims 1-9; figures 1-14 -----	1-12
X	US 2002/091720 A1 (LIU JUN [US] ET AL) 11 July 2002 (2002-07-11) abstract paragraph [0003] - paragraph [0008] paragraph [0019] - paragraph [0031] paragraph [0033] - paragraph [0040] paragraph [0044] - paragraph [0048] claims 1-24; figures 1-5 -----	1-12
X	PETER PAWLAK: "Software Update Service to Ease Patch Distribution", INTERNET CITATION, 12 November 2008 (2008-11-12), pages 1-6, XP002503977, Retrieved from the Internet: URL:http://www.directionsonmicrosoft.com/sample/DOMIS/update/2002/05may/0502sustep.htm [retrieved on 2008-11-12] page 1, paragraph 1 page 1, paragraph 5 - page 2, paragraph 3 page 2, paragraph 6 - paragraph 7 page 2, paragraph 9 - page 4, paragraph 8 page 4, paragraph 10 - page 6, paragraph 3 -----	1-12
X	US 2005/289536 A1 (NAYAK RANJIT [US] ET AL) 29 December 2005 (2005-12-29) abstract paragraph [0001] - paragraph [0009] paragraph [0012] - paragraph [0015] paragraph [0024] - paragraph [0036] paragraph [0038] - paragraph [0052] paragraph [0053] - paragraph [0068] claims 1-29; figures 1-6 ----- -/--	1-12

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2011/050527

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 2005/132349 A1 (ROBERTS JASON [US] ET AL) 16 June 2005 (2005-06-16) abstract paragraph [0002] - paragraph [0009] paragraph [0013] - paragraph [0020] paragraph [0037] - paragraph [0055] paragraph [0057] - paragraph [0077] paragraph [0079] - paragraph [0102] paragraph [0105] - paragraph [0108] claims 1-25; figures 1-13 -----	1-12
X	US 2007/033652 A1 (SHERWANI ADIL A [US] ET AL) 8 February 2007 (2007-02-08) abstract paragraph [0005] - paragraph [0006] paragraph [0020] - paragraph [0034] paragraph [0037] - paragraph [0046] paragraph [0050] - paragraph [0052] paragraph [0059] - paragraph [0098] claims 1-18; figures 1-11 -----	1-12
X	US 7 155 490 B1 (MALMER FREDRIK [SE] ET AL) 26 December 2006 (2006-12-26) abstract column 2, line 32 - column 3, line 16 column 3, line 46 - column 4, line 16 column 5, line 39 - column 6, line 67 column 7, line 8 - column 9, line 4 column 9, line 30 - column 14, line 13 column 14, line 29 - line 49 claims 1-21; figures 1-6 -----	1-12
A	EP 1 672 490 A2 (MICROSOFT CORP [US]) 21 June 2006 (2006-06-21) abstract paragraph [0002] - paragraph [0012] paragraph [0014] - paragraph [0023] paragraph [0026] - paragraph [0035] paragraph [0037] - paragraph [0046] paragraph [0048] - paragraph [0049] claims 1-32; figures 1-8 -----	1-12
A	MICROSOFT CORPORATION: "Deploying Microsoft Software Update Services", INTERNET CITATION, 5 March 2007 (2007-03-05), 5 March 2007 (2007-03-05), XP002423137, Retrieved from the Internet: URL:http://web.archive.org/we/20051013100250/http://download.microsoft.com/download/c/d/8/cd8ac959-37a8-4e5b-860a-465b179984af/SUS_Deployguide _sp1.doc [retrieved on 2007-03-05] the whole document ----- -/--	1-12

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2011/050527

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 00/58855 A1 (QUARK INC [US]; QUARK MEDIA HOUSE SARL [CH]) 5 October 2000 (2000-10-05) the whole document -----	1-12

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/EP2011/050527

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2007118617 A1	24-05-2007	CA 2630851 A1	25-10-2007
		EP 1952616 A2	06-08-2008
		JP 2009517926 T	30-04-2009
		KR 20080071195 A	01-08-2008
		US 2011072146 A1	24-03-2011
		WO 2007120288 A2	25-10-2007
EP 1832976 A1	12-09-2007	WO 2006040810 A1	20-04-2006
		US 2007244999 A1	18-10-2007
US 2002091720 A1	11-07-2002	NONE	
US 2005289536 A1	29-12-2005	NONE	
US 2005132349 A1	16-06-2005	AU 2004279162 A1	30-06-2005
		BR PI0406412 A	04-10-2005
		CA 2501499 A1	15-06-2005
		CN 101410800 A	15-04-2009
		EP 1579301 A2	28-09-2005
		JP 2007523395 T	16-08-2007
		KR 20060109284 A	19-10-2006
		MX PA05006615 A	21-10-2005
		RU 2365983 C2	27-08-2009
		WO 2005060387 A2	07-07-2005
US 2007033652 A1	08-02-2007	AU 2006270273 A1	25-01-2007
		BR PI0612865 A2	30-11-2010
		CA 2612631 A1	25-01-2007
		CN 101310472 A	19-11-2008
		EP 1905184 A2	02-04-2008
		JP 2009503648 T	29-01-2009
		KR 20080026132 A	24-03-2008
		WO 2007011667 A2	25-01-2007
US 7155490 B1	26-12-2006	US 2009204954 A1	13-08-2009
		US 2007106757 A1	10-05-2007
EP 1672490 A2	21-06-2006	AU 2005239708 A1	29-06-2006
		BR PI0505512 A	12-09-2006
		CA 2529966 A1	14-06-2006
		CN 1790266 A	21-06-2006
		JP 2006172472 A	29-06-2006
		KR 20060067842 A	20-06-2006
		MX PA05013575 A	15-06-2006
		US 2006130037 A1	15-06-2006
WO 0058855 A1	05-10-2000	AU 4181400 A	16-10-2000
		CA 2367889 A1	05-10-2000
		EP 1171830 A1	16-01-2002
		JP 2002540529 T	26-11-2002