

(19) World Intellectual Property
Organization
International Bureau



(43) International Publication Date
8 December 2005 (08.12.2005)

PCT

(10) International Publication Number
WO 2005/116836 A1

(51) International Patent Classification⁷: **G06F 11/34**

(21) International Application Number:
PCT/US2005/016995

(22) International Filing Date: 17 May 2005 (17.05.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/571,774 17 May 2004 (17.05.2004) US

(71) Applicant (for all designated States except US): **COM-
PUTER ASSOCIATES THINK, INC.** [US/US]; One
Corporate Associates Plaza, Islandia, NY 11749 (US).

(72) Inventor; and

(75) Inventor/Applicant (for US only): **WENGER, Joseph**
[US/US]; 1611 Hyland Greens Drive, Grapevine, TX
76051 (US).

(74) Agent: **BHAVSAR, Samir, A.**; Baker Botts L.L.P., 2001
Ross Avenue, Dallas, TX 75201 (US).

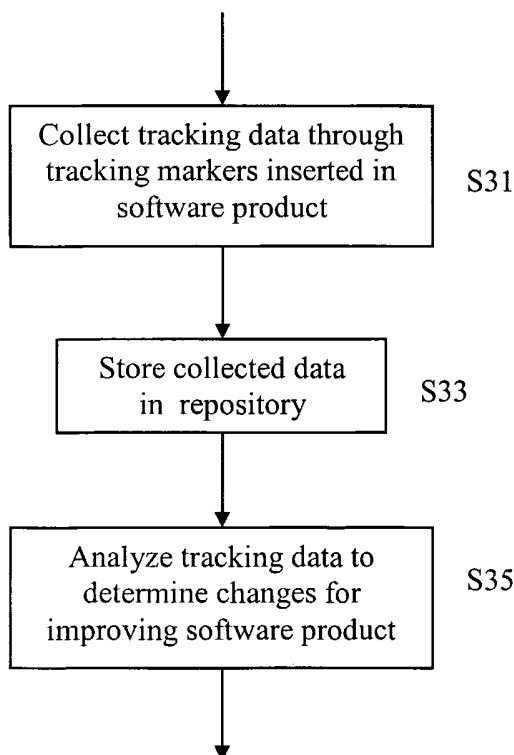
(81) Designated States (unless otherwise indicated, for every
kind of national protection available): AE, AG, AL, AM,
AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN,
CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI,
GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE,
KG, KM, KP, KR, KZ, LC, LK,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

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

Published:
— with international search report

[Continued on next page]

(54) Title: METHOD AND APPARATUS FOR IMPROVING A SOFTWARE PRODUCT



(57) Abstract: Methods and apparatuses for improving a software product are provided. A method for improving a software product includes collecting tracking data (531) through tracking markers inserted at respective functional points of the software product, storing the collected tracking data in a repository (533), and retrieving the tracking data from the repository and analyzing the tracking data to determine changes for improving the software product (535).

WO 2005/116836 A1



-
- *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*
- For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.*

METHOD AND APPARATUS FOR IMPROVING A SOFTWARE PRODUCT**TECHNICAL FIELD**

The present disclosure relates to product
5 performance and utilization and, more specifically, to
techniques for improving a software product by tracking
product performance and utilization trends.

DESCRIPTION OF THE RELATED ART

10 In the endeavor to continuously improve a product
(for example, a computer software product), it is
increasingly important to obtain accurate feedback as to
performance of the product.

One approach to product improvement is to conduct
15 product testing at the vendor site. While vendor-site
testing may be a useful approach to testing the
effectiveness of a product, many of the issues faced by
clients at client-sites may not be brought to light
during vendor-site testing.

20 Another approach is to employ beta testing. In beta
testing, a pre-release version of a product is released
to a limited number of beta testers. The product may
then be utilized by the beta testers at the beta test-
sites. Detailed feedback may then be provided from the

beta testers to the vendor which may help for correction of problems and development of improvements, prior to general release of the product. Beta testing may continue after the product has been released, to obtain
5 additional information that may be used to improve the product and/or subsequent versions of the product.

However, a drawback of beta testing is that the scope of the testing is typically limited to a small set of beta testers. Broad based testing has generally been
10 impracticable due to such factors as unacceptable costs and lack of detailed feedback from clients.

Client feedback, often in the form of trouble areas and/or suggested improvements, can be a useful tool in product improvement. However, there is a need for a
15 broader base of detailed feedback. This feedback may then be used to develop product improvements that are more responsive to the needs of the clients.

SUMMARY

20 This application describes methods, apparatuses and systems for improving a software product.

A method for improving a software product, according to one exemplary embodiment, includes activating tracking markers inserted at respective functional points of the

software product, collecting through the tracking markers tracking data corresponding to operation of the software product, during the operation of the software product, using the collected tracking data to generate activity
5 records, writing the activity records corresponding to the collected tracking data to a repository, and sending the activity records from the repository to a destination for determining improvements to the software product.

According to another exemplary embodiment, a method
10 for improving a software product includes collecting tracking data through tracking markers inserted at respective functional points of the software product, storing the collected tracking data in a repository, and retrieving the tracking data from the repository and
15 analyzing the tracking data to determine changes for improving the software product.

An apparatus for improving a software product, according to one exemplary embodiment, includes a repository, one or more tracking markers, and a tracking
20 data analyzer. The tracking markers are inserted at one or more functional points within the software product, to collect tracking data while the software product is in operation, and deposit the collected tracking data into the repository. The tracking data analyzer analyzes the

collected tracking data to determine performance and utilization trends.

The application also describes methods, apparatuses and systems for assessing product performance.

5 A method for assessing product performance of a software product, according to one exemplary embodiment, includes collecting performance data at one or more functional points within the software product, storing the collected performance data in a central repository,
10 and communicating the stored data to a location where the stored data is analyzed.

 An apparatus for assessing product performance of a software product, according to one exemplary embodiment, includes one or more markers for collecting performance
15 data at one or more functional points within the software product, and a central repository for storing the collected performance data. The stored data can be communicated over a network to a location where the stored data is analyzed.

20 The methods, apparatuses and systems of this disclosure may be embodied in one or more computer programs stored on a computer readable medium or program storage device and/or transmitted via a computer network or other transmission medium in one or more segments or

packets.

BRIEF DESCRIPTION OF THE DRAWINGS

A more complete appreciation of the present
5 disclosure and many of the attendant advantages thereof
will be readily obtained as the same becomes better
understood by reference to the following detailed
description when considered in connection with the
accompanying drawings, wherein:

10 Fig. 1 shows a block diagram of an exemplary
computer system capable of implementing the methods and
apparatuses of the present disclosure;

Fig. 2 shows a block diagram illustrating an
apparatus for improving a software product, according to
15 an exemplary embodiment of the present disclosure;

Fig. 3 shows a flow chart illustrating a method for
improving a software product, according to the embodiment
of FIG. 2;

Fig. 4 shows a flow chart illustrating a method for
20 improving a software product, according to another
embodiment;

Fig. 5 is a block diagram showing a system for
collecting data, according to an exemplary embodiment of
the present disclosure;

FIG. 6 is a schematic diagram showing an example of an exemplary activity record; and

FIG. 7 is a flow chart illustrating a method for collecting and analyzing data according to an exemplary
5 embodiment.

DETAILED DESCRIPTION

The present disclosure provides tools (in the form of methodologies, apparatuses, and systems) for improving
10 a software product.

The following exemplary embodiments are set forth to aid in an understanding of the subject matter of this disclosure, but are not intended, and should not be construed, to limit in any way the claims which follow
15 thereafter. Therefore, while specific terminology is employed for the sake of clarity in describing some exemplary embodiments, the present disclosure is not intended to be limited to the specific terminology so selected, and it is to be understood that each specific
20 element includes all technical equivalents which operate in a similar manner.

FIG. 1 shows an example of a computer system 100 which can implement the methods and apparatuses of the present disclosure. The apparatuses and methods of the

present disclosure may be implemented in the form of a software application running on a computer system, for example, a mainframe, personal computer (PC), handheld computer, server, etc. The software application may be
5 stored on a recording media locally accessible by the computer system, for example, floppy disk, compact disk, hard disk, etc., or may be remote from the computer system and accessible via a hard wired or wireless connection to a computer network, (for example, a local
10 area network, the Internet, etc.) or another transmission medium. Alternatively, the apparatuses and methods of this application, as will be apparent to one skilled in the art after reading this disclosure, can be implemented in hardware or firmware.

15 The computer system 100 can include a central processing unit (CPU) 102, program and data storage devices 104, a printer interface 106, a display unit 108, a (LAN) local area network data transmission controller 110, a LAN interface 112, a network controller 114, an
20 internal bus 116, and one or more input devices 118 (for example, a keyboard, mouse etc.). As shown, the system 100 may be connected to a database 120, via a link 122.

Techniques for improving a software product are discussed below.

An exemplary embodiment is discussed below with reference to Figs. 2 and 3. An apparatus 20 for improving a software product, according to an, includes a repository 21, one or more tracking markers 22, and a tracking data analyzer 23. The tracking markers 22 are inserted at one or more functional points within the software product 25, to collect tracking data while the software product is in operation, and deposit the collected tracking data into the repository 21. The tracking data analyzer 23 analyzes the collected tracking data to determine performance and utilization trends.

A tracking marker 22 can be a module configured to collect predetermined types of data during operation of the software module, and collection of tracking data by the tracking marker can be specified by logic included in the software product, for the tracking marker, and/or initiated by user action. The tracking markers can operate continuously to collect the tracking data.

The tracking data stored in repository 21 can include product performance data, product activity data, utilization data, etc.

The tracking data analyzer 23 can analyze the collected tracking data to acquire performance and

utilization statistics, and/or determine improvements for the software product

A method for improving a software product can include collecting tracking data through the tracking
5 markers 22 inserted at respective functional points of the software product 25 (step **S31**), storing the collected tracking data in the repository 21 (step **S33**), and retrieving the tracking data from the repository 21 and analyzing the tracking data with the analyzer 23 to
10 determine changes for improving the software product 25 (step **S35**).

Collection of tracking data by a tracking marker can be automated, and/or specified by logic included in the software product, for the tracking marker.

15 The method may further include (a) activating the tracking markers, wherein the activation of a tracking marker is specified by logic included in the software product, for the tracking marker, and/or triggered by user action, (b) analyzing the tracking data to acquire
20 performance and utilization statistics, and the performance and utilization statistics are used to improve the software product, (c) communicating the tracking data to a product vendor over a network prior to analysis of the tracking data, wherein the analysis of

the tracking data is performed by the product vendor, and/or (d) generating one or more activity records by using the collected tracking data, and storing the one or more activity records in the repository.

5 A method for improving a software product, according to another exemplary embodiment (Fig. 4), includes activating tracking markers inserted at respective functional points of the software product (step **S41**), collecting through the tracking markers tracking data
10 corresponding to operation of the software product during the operation of the software product (step **S43**), using the collected tracking data to generate activity records (step **S45**), writing the activity records corresponding to the collected tracking data to a repository (step **S47**),
15 and sending the activity records from the repository to a destination for determining improvements to the software product (step **S49**). The method may further include analyzing statistically a plurality of the activity records to determine performance and utilization trends.

20 The software product can be host system software and the tracking marker collects host system performance data. The activity record containing the host system performance data can include host system identifying information identifying the host system software.

Activity records can include (i) date and timestamp information identifying the date and time at which corresponding tracking data was collected, (ii) marker location information identifying a location in the software product of a tracking marker which collected the
5 corresponding tracking data of the activity record, and/or (iii) a marker description describing the tracker marker which collected the corresponding tracking data of the activity record. In addition, activity records can
10 optionally include product error statistics, input/output statistics, storage utilization information, as well as various other types of information. The activity information in an activity record can describes software execution activity during code execution at the
15 corresponding functional point in the software product.

A method for assessing product performance, according to an exemplary embodiment, includes collecting performance data at one or more functional points within the product, storing the collected performance data in a
20 central repository, and communicating the stored data to a location where the stored data may be analyzed.

The collection of the performance data can be performed by one or more markers inserted at one or more functional points within the product. Collection of

performance data can be initiated automatically, by a user, by logic incorporated into the software product, and/or by a product vendor, and can be performed continuously.

5 The collected data can comprise activity data, identification data, tracking data, etc., can be stored in one or more activity records in the repository and/or in a compressed format, can be communicated to a product vendor (for example, over a wide area network, a local
10 area network, the Internet, etc., and via FTP or email), and can be analyzed by the product vendor (or another) for performance and utilization statistics and/or for improvements to the product.

 An apparatus for assessing product performance,
15 according to an exemplary embodiment, includes one or more markers for collecting performance data at one or more functional points within the product, a central repository for storing the collected performance data, and a communications network for communicating the stored
20 data to a location where the stored data may be analyzed.

 The apparatus can additionally include an analyzing unit for analyzing the collected data, for example, to determine performance and/or utilization trends, and/or improvements to the product.

As indicated above, one of the steps in the process for improving a product is to collect, preferably using an automated process, a wide range of data pertaining to performance and utilization of the product at client-
5 sites. As discussed herein, tracking markers can be inserted at functional points in the product, and used to collect tracking data (for example, performance and utilization data). Collected data may then be forwarded to the vendor for analysis and/or review, or
10 automatically (and/or periodically) analyzed using a smart analyzer.

Tracking data may relate to, for example, utilization review, capacity planning and/or functional flow tracking in a product. Tracking data may include
15 information pertaining to a marker as established during the placement of the marker into the product. Tracking data may include host system level information indicating host system performance placement of product {ask inventor what this means}. Tracking data may include
20 host system identification information (for example, the operating system in use by the host system and other information that identifies the platform of the host system). Tracking data may include statistics relating to errors that are encountered during operation of the

product. Tracking data may include input/output (I/O) statistics. Tracking data may include storage utilization information. Tracking data may include a wide range of other data that may be pertinent to the performance of the product (for example, information that
5 pertains to a particular marker location as described below).

As another example, tracking data can include activity data that describes what is actually occurring
10 during operation of the product at specific points.

The collected data can also include identification information. The identification information can include information describing data that has been collected, such that collected data can properly (and quickly) be
15 identified. The identification information can also identify the client site, for example, by specifying a particular client and/or a particular client facility where the product is utilized. The identification information can include client host system
20 identification, such as an indicator specifying a particular system in which the product is installed. The identification information can include product identification, for example, specifying the particular product for or from which data is being collected. Since

a client may have multiple products from a particular vendor installed on their systems, data relating to multiple products may be collected concurrently. The identification information can also specify release level
5 (for example, the particular version and/or update/patch implementation of the product) and/or maintenance applied (for example, patches and/or updates that have been applied to the product, customizations and settings applied to the product, etc.). The identification
10 information can include, in addition, module/program identification (for example, specifying a particular module and/or program within a product to which collected data relates), marker identification (for example, specifying the identity of the particular marker that has
15 collected the data), and module/program marker location (for example, identifying a particular location within a particular module or program where a marker has been placed, and hence, from where the data has been collected). The identification information can include
20 date and time stamp of marker activity (for example, indicating the date and time at which the marker collected the data). The identification information can also include a wide range of other indicators that may contribute to identifying a context for collected data.

According to some embodiments, logic for initializing monitoring of product processes may be incorporated into the product. Tracking markers may be placed at specific functional points within the product.

5 Fig. 5 is a block diagram showing a system for collecting data, according to an exemplary embodiment. Product 50 (for example, a software product) may contain multiple functional points within the product where events or consequence occur. For example, functional
10 points may be inserted in modules, objects, functions, processes, etc., within the product. On the other hand, functional points may be events that occur during product use. Tracking markers 53 may be placed at one or more of the multiple functional points.

15 The markers 53 may be objects, program elements, units of code, processes, daemons, etc., that monitor functional points to collect pertinent data, for example, tracking data, activity data, etc. The markers may monitor data continuously, intermittently or when
20 triggered to do so, and may collect data automatically and/or upon request from the client and/or the vendor. The markers may collect data actively or passively. For example, the markers may record data during operation of the product (such as execution time). As another

example, a marker may actively ask a user for user feedback.

Markers can send collected data (for example, tracking data, activity data, etc.) to the repository 55.

5 The repository may be at the client-site. A single repository may be used for each client, for each client-site, for each vendor and/or for each product. Collected data may be stored in one or more external files within the repository. Collected data may be stored as a series

10 of activity records 56. For example, single activity record may be stored for each marker for each time the marker records data. Alternatively, data may be stored in the repository in another form (for example, optimized for statistically analyzing the stored data). Fig. 6

15 shows schematically an example of an activity record according to an exemplary embodiment.

Stored data may be encapsulated and/or compressed for efficient storage and/or communication. Additionally, data stored in the repository 55 may be

20 accessible to the client for the client's information and/or use.

In addition to storing collected data, the repository 55 (or another repository) may be used to store one or more control files. The control files may

be used to store and relay operational options, for example, options concerning the collection and communication of data.

Collected data may then be recovered by vendor 58
5 for analysis. Various methods may be used to allow the vendor to recover collected data. Collected data, for example, in the form of activity records 56, may be periodically sent to the vendor, such as over a network 57, via email or via FTP. Alternatively, the activity
10 records may be transmitted over a virtual private network or an alternative form of secure connection. According to another embodiment, the activity records may be sent to the vendor 58 across a local area network or wide area network.

15 Collected data may be "pushed" to the vendor as it is collected (for example, transfer of data is initiated at the client-site). Email or FTP may be used to push collected data. In some instances, collected data may be pushed when a certain volume of data has been collected.

20 According to another embodiment, collected data may be "pulled" to the vendor at the vendor's request (for example, transfer of data is initiated at the vendor-site). In some instances, collected data may be pulled when the vendor is interested in perfuming analysis on

the data.

The data may be received by a development module and/or group on the vendor side. Where multiple copies of multiple products are implemented at client-sites
5 world-wide, data from multiple client-sites may be received, for example, at a single site. The data may then be analyzed to determine how the product functions in the client-site. In conducting the analysis, the product's response to the client's environment may be
10 clearly understood. Additionally, utilization review, capacity planning and functional flow tracking may all be performed on the recovered data.

Tracking markers may be embedded at various functional points within the product during compilation.
15 Alternatively, the markers may be inserted at the functional points during operation of the product. The product may then be installed at the client-site. The markers may contain logic appropriate for the process or location that the vendor has a particular interest in
20 monitoring, for example for later development.

Fig. 7 is a flow chart illustrating a method for collecting and analyzing data according to an embodiment of the present disclosure. The markers may be initialized (Step S71). Initialization of markers

activates the data collection process. This step may be executed, for example, by a predefined schedule, periodically, manually (for example by the client or by the vendor) and/or according to volume (for example based
5 on a predefined number of transactions). Predefined values and other adjustable parameters may be read from the control file. Once markers are initialized, data collection may begin.

Collected data may then be written to files, for
10 example activity records (Step S72), for example, within the repository. When data is to be recovered, for example as described above, the data may be communicated to the vendor (Step S73), for example as described above.

The vendor may then have the data analyzed (Step
15 S74), for example as described above. Analyzed data may then be utilized (Step S75) to improve the product or subsequent versions of the product.

The specific embodiments discussed herein are illustrative, and many additional modifications and
20 variations can be introduced on these embodiments without departing from the spirit of the disclosure or from the scope of the appended claims. For example, elements (such as steps) and/or features of different illustrative embodiments may be combined with each other and/or

substituted for each other within the scope of this disclosure and appended claims.

Additional variations may be apparent to one of ordinary skill in the art from reading U.S. provisional
5 application Serial No. 60/571,774, filed May 21, 2004, the entire contents of which are incorporated herein by reference.

What is claimed is:

1. A method for improving a software product, said method comprising:

activating tracking markers inserted at respective
5 functional points of the software product;

collecting through the tracking markers tracking data corresponding to operation of the software product, during the operation of the software product;

using the collected tracking data to generate
10 activity records;

writing the activity records corresponding to the collected tracking data to a repository; and

sending the activity records from the repository to a destination for determining improvements to the
15 software product.

2. The method of claim 1, further comprising analyzing statistically a plurality of the activity records to determine performance and utilization trends.

20

3. The method of claim 1, wherein the tracking marker is a module configured to collect predetermined types of data during operation of the software module.

4. The method of claim 1, wherein the software product is host system software and the tracking marker collects host system performance data.

5 5. The method of claim 4, wherein the activity record containing the host system performance data also includes host system identifying information identifying the host system software.

10 6. The method of claim 1, wherein each activity record includes date and timestamp information identifying the date and time at which corresponding tracking data was collected.

15 7. The method of claim 1, wherein each activity record includes marker location information identifying a location in the software product of a tracking marker which collected the corresponding tracking data of the activity record.

20

8. The method of claim 1, wherein each activity record includes a marker description describing the tracker marker which collected the corresponding tracking data of the activity record.

9. The method of claim 1, wherein at least one of the activity records includes product error statistics.

5 10. The method of claim 1, wherein at least one of the activity records includes input/output statistics.

11. The method of claim 1, wherein at least one of the activity records includes storage utilization
10 information.

12. The method of claim 1, wherein the activity information in at least one of the activity records describes software execution activity during code
15 execution at the corresponding functional point in the software product.

13. A computer system, comprising:
a processor; and
20 a program storage device readable by the computer system, tangibly embodying a program of instructions executable by the processor to perform the method claimed in claim 1.

14. A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform the method claimed in claim 1.

5 15. A computer data signal transmitted in one or more segments in a transmission medium which embodies instructions executable by a computer to perform the method claimed in claim 1.

10 16. A method for improving a software product, said method comprising the steps of:

(a) collecting tracking data through tracking markers inserted at respective functional points of the software product, said tracking data being collected
15 while the software product is in operation;

(b) storing the tracking data collected in step (a) in a repository; and

(c) retrieving the tracking data from the repository and analyzing the tracking data to determine changes for
20 improving the software product.

17. The method of claim 16, wherein collection of the tracking data in step (a) is automated.

18. The method of claim 16, wherein collection of tracking data by a tracking marker is specified by logic included in the software product, for the tracking marker.

5

19. The method of claim 16, further comprising activating the tracking markers prior to step (a), wherein the activation of a tracking marker is specified by logic included in the software product, for
10 the tracking marker.

20. The method of claim 16, further comprising activating the tracking markers prior to step (a), wherein the activation of a tracking marker is
15 triggered by user action.

21. The method of claim 16, further comprising analyzing the tracking data to acquire performance and utilization statistics, and the performance and
20 utilization statistics are used to improve the software product in step (c).

22. The method of claim 16, further comprising communicating the tracking data to a product vendor

over a network prior to analysis of the tracking data in step (c),

wherein the analysis of the tracking data in step (c) is performed by the product vendor.

5

23. The method of claim 16, further comprising generating one or more activity records by using the tracking data collected in step (a), and

storing the one or more activity records in the repository.

10

24. A computer system, comprising:

a processor; and

a program storage device readable by the computer system, tangibly embodying a program of instructions executable by the processor to perform the method claimed in claim 16.

15

25. A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform the method claimed in claim 16.

20

26. A computer data signal transmitted in one or more segments in a transmission medium which embodies

instructions executable by a computer to perform the method claimed in claim 16.

27. A apparatus for improving a software product,
5 said apparatus comprising:

a repository;

one or more tracking markers inserted at one or more functional points within the software product, to collect tracking data while the software product is in operation,
10 and deposit the collected tracking data into the repository; and

a tracking data analyzer, wherein the tracking data analyzer analyzes the collected tracking data to determine performance and utilization trends.

15

28. The apparatus of claim 27, wherein the tracking marker is a module configured to collect predetermined types of data during operation of the software module.

20 29. The apparatus of claim 27, wherein collection of tracking data by a tracking marker is specified by logic included in the software product, for the tracking marker.

30. The apparatus of claim 27, wherein the collection of the tracking data by the tracking markers is initiated by user action.

5 31. The apparatus of claim 27, wherein the tracking markers operate continuously to collect the tracking data.

32. The apparatus of claim 27, wherein the
10 collected tracking data includes product performance data.

33. The apparatus of claim 27, wherein the collected tracking data includes product activity data.

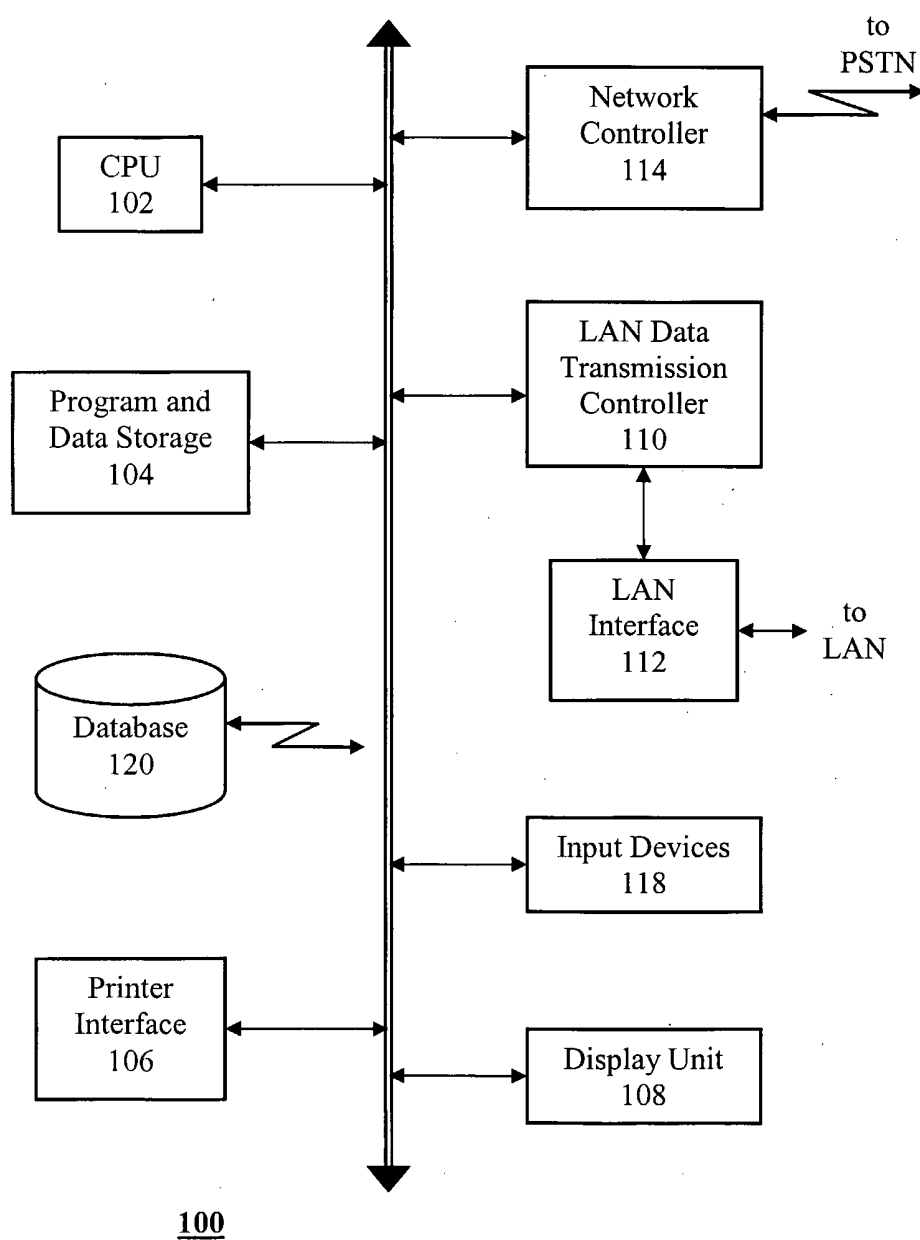
15

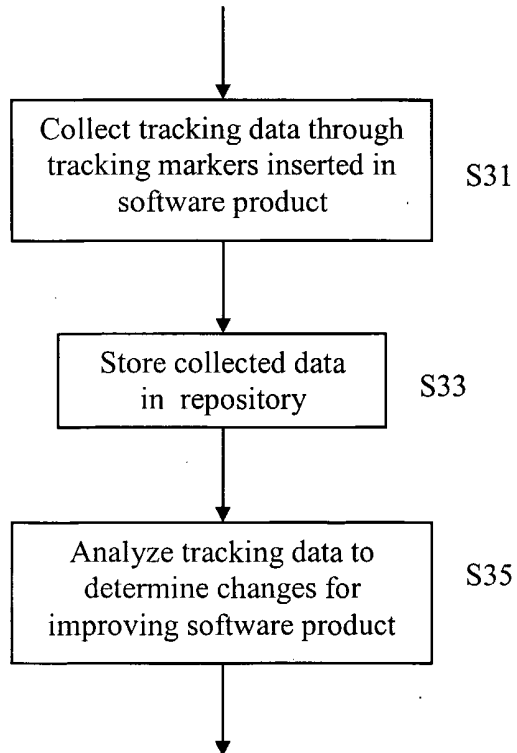
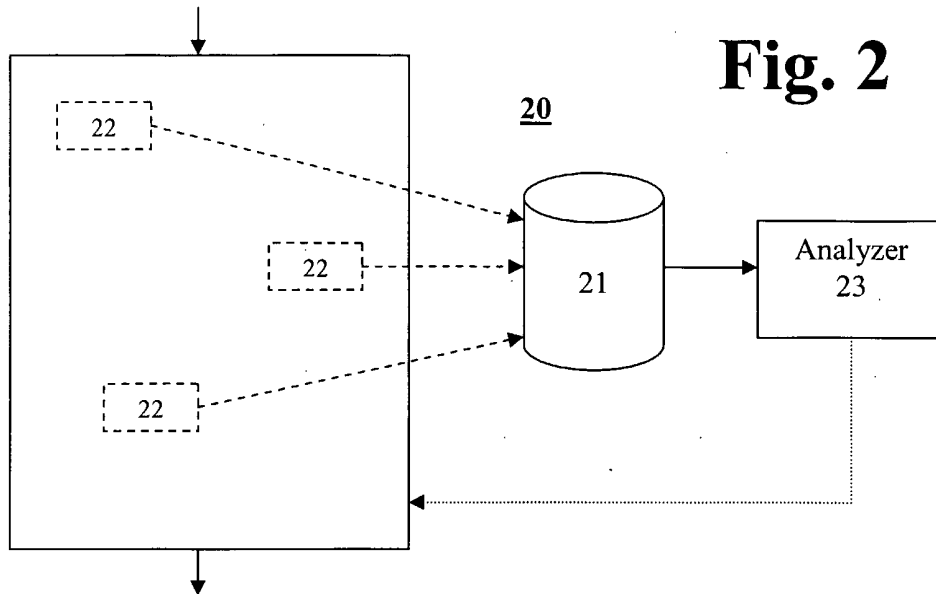
34. The apparatus of claim 27, wherein the collected tracking data includes utilization data.

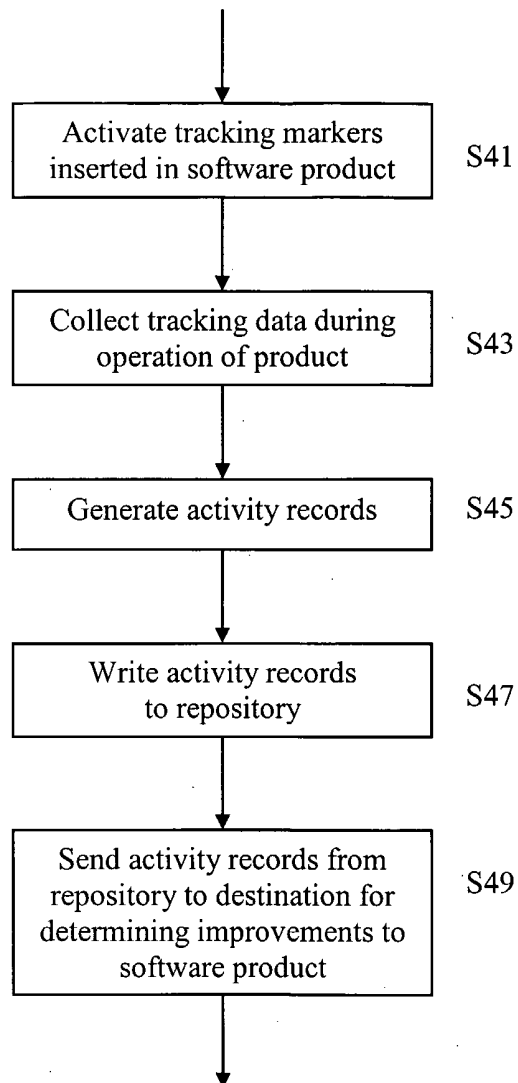
35. The apparatus of claim 27, wherein said tracking
20 data analyzer analyzes the collected tracking data to acquire performance and utilization statistics.

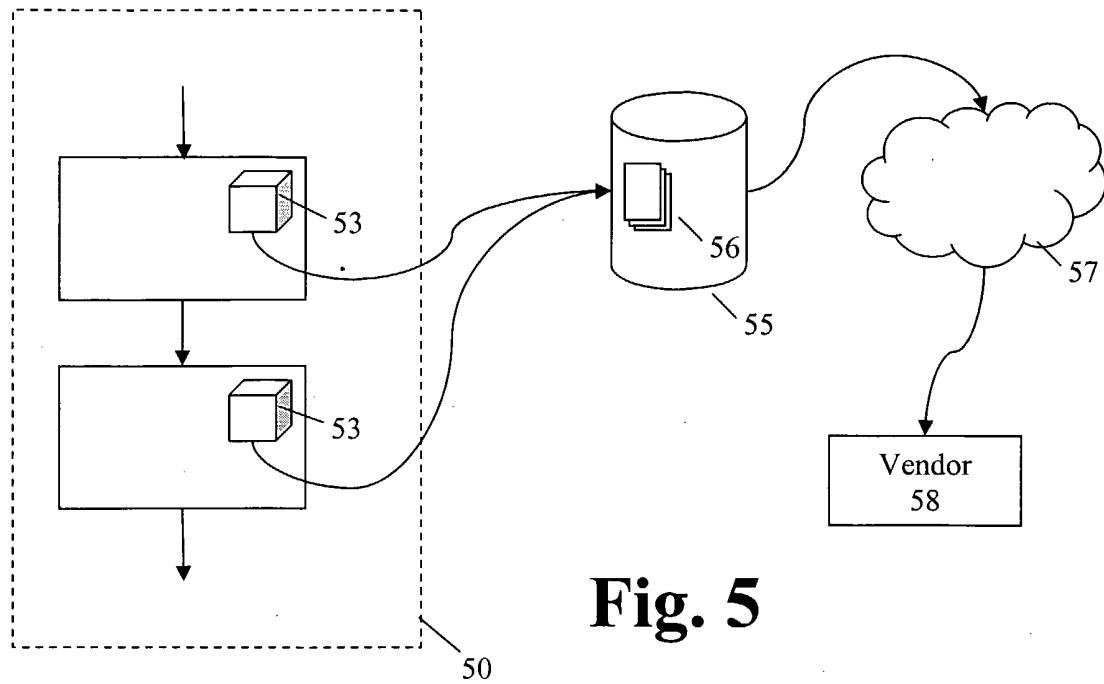
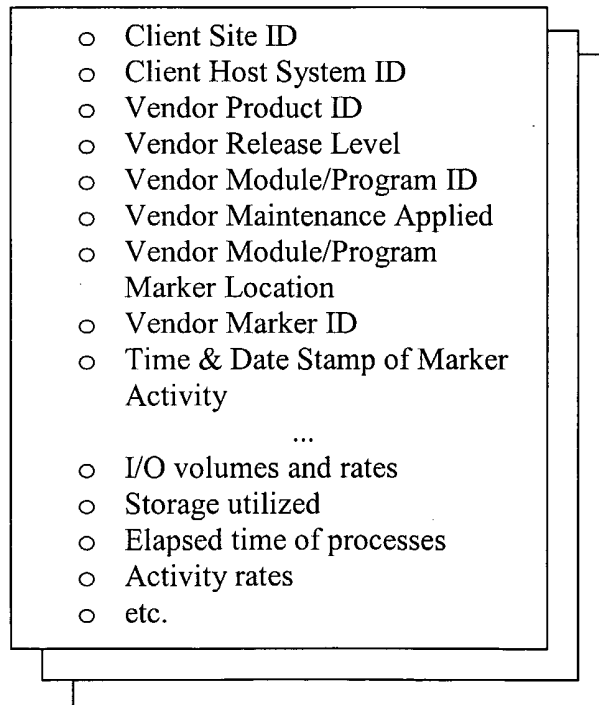
36. The apparatus of claim 27, wherein said tracking data analyzer analyzes said communicated data to

determine improvements for the software product.

Fig. 1



**Fig. 4**

**Fig. 5****Fig. 6**

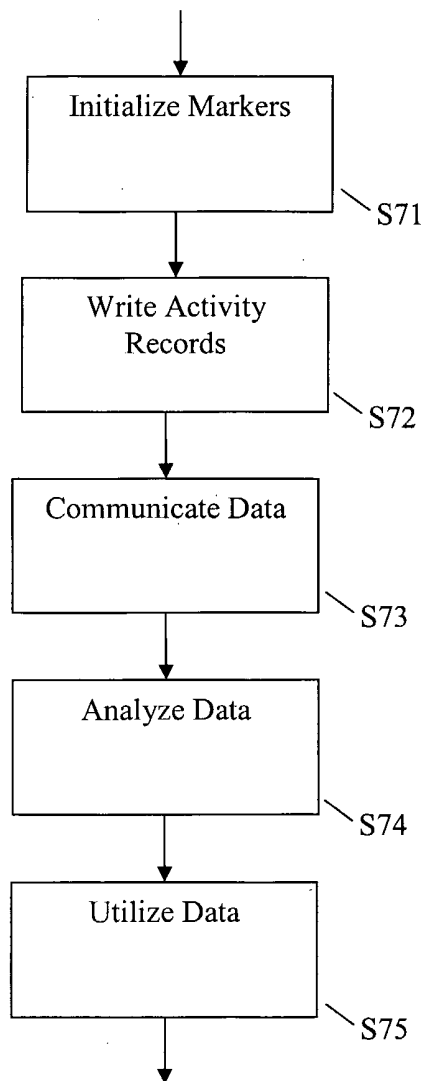


Fig. 7

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US2005/016995

A. CLASSIFICATION OF SUBJECT MATTER

IPC 7 G06F11/34

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)

IPC 7 G06F

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

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

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 918 004 A (ANDERSON ET AL) 29 June 1999 (1999-06-29) abstract column 1, line 35 - line 42 column 1, line 62 - column 2, line 12 column 4, line 3 - line 51 column 5, line 3 - line 44 column 6, line 49 - line 64 column 9, line 1 - line 4 -----	1-36
X	US 6 126 330 A (KNIGHT ET AL) 3 October 2000 (2000-10-03) abstract column 1, line 16 - line 57 column 3, line 11 - line 22 column 5, line 25 - line 48 column 6, line 22 - column 7, line 9 column 7, line 52 - column 8, line 38 ----- -/--	1-36

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

☒ Patent family members are listed in 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

21 September 2005

Date of mailing of the international search report

30/09/2005

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Leuridan, K

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US2005/016995

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 00/72151 A (KONINKLIJKE PHILIPS ELECTRONICS N.V) 30 November 2000 (2000-11-30) the whole document -----	1-36
X	US 5 828 883 A (HALL ET AL) 27 October 1998 (1998-10-27) abstract column 2, line 5 - line 8 column 2, line 61 - column 3, line 6 column 9, line 34 - line 64 -----	1,2,4, 10,11, 13-17, 21, 23-27, 32-36
X	WO 95/27249 A (INTEL CORPORATION) 12 October 1995 (1995-10-12) abstract -----	1,13-16, 24-27
X	US 6 189 142 B1 (JOHNSTON GARY MARTIN ET AL) 13 February 2001 (2001-02-13) abstract -----	1,13-16, 24-27
A	"DEKOVERT V PROGRAM FOR OS/2 MEMORY-MAPPED PERFORMANCE HOOKS" IBM TECHNICAL DISCLOSURE BULLETIN, IBM CORP. NEW YORK, US, vol. 35, no. 5, 1 October 1992 (1992-10-01), pages 150-152, XP000312915 ISSN: 0018-8689 the whole document -----	1-36

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/US2005/016995

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
US 5918004	A	29-06-1999	CA 2224689 A1 EP 0830611 A1 WO 9638733 A1	05-12-1996 25-03-1998 05-12-1996
US 6126330	A	03-10-2000	NONE	
WO 0072151	A	30-11-2000	US 6374369 B1	16-04-2002
US 5828883	A	27-10-1998	NONE	
WO 9527249	A	12-10-1995	AU 2237995 A CN 1516012 A CN 1149343 A DE 69518745 D1 DE 69518745 T2 EP 0754321 A1 JP 10501907 T	23-10-1995 28-07-2004 07-05-1997 12-10-2000 12-07-2001 22-01-1997 17-02-1998
US 6189142	B1	13-02-2001	NONE	