

SEARCH:

[GO TO ADVANCED SEARCH](#)

LOGGED IN AS:

- [glen petrie](#)
- [Logout](#)
- [HOME](#)
- [SEARCH PATENTS](#)
- [CHEMICAL SEARCH](#)
- [DATA SERVICES](#)
- [HELP](#)
- [My Account](#)
- [My Portfolios](#)
- [My Alerts](#)
- [My Saved Searches](#)
- [Invite a Friend](#)

Portfolio:

Add to portfolio or add to a new portfolio, named



[Plaque It!](#)

Title:

IMAGE PROCESSING APPARATUS, LOG RECORDING METHOD, AND STORAGE MEDIUM

Document Type and Number:

United States Patent Application 20080112009

Kind Code:

A1

Abstract:

An image processing apparatus where an application level function is implemented by one or more basic functions is disclosed. The disclosed image processing apparatus includes a log recording unit configured to record basic function logs of the basic functions according to log formats of the basic functions, the basic function logs being associated with an application level function log of the application level function.

Inventors:
Tojo, Yoshiharu (Tokyo, JP)

Application Number:
11/936408

Publication Date:
05/15/2008

Filing Date:
11/07/2007

View Patent Images:

[20080112009](#)  Referenced by:
[View patents that cite this patent](#)

[PDF help](#)

Export Citation:

[Click for automatic bibliography generation](#)

Primary Class:

[358/1.15](#)

International Classes:

G06F15/00

Attorney, Agent or Firm:

OBLON, SPIVAK, MCCLELLAND MAIER & NEUSTADT, P.C. (1940 DUKE STREET, ALEXANDRIA, VA, 22314, US)

Claims:

What is claimed is:

1. An image processing apparatus where an application level function is implemented by one or more basic functions, comprising: a log recording

unit configured to record basic function logs of the basic functions according to log formats of the basic functions, the basic function logs being associated with an application level function log of the application level function.

2. The image processing apparatus as claimed in claim 1, wherein the log recording unit is configured to attach an image processing ID identifying an image processing job performed by the application level function to the application level function log and to each of the basic function logs, the basic function logs being associated with the application level function log by the image processing ID.

3. The image processing apparatus as claimed in claim 1, wherein the log recording unit is configured to store pointers to the basic function logs in the application level function log, the basic function logs being associated with the application level function log by the pointers.

4. The image processing apparatus as claimed in claim 1, further comprising: a framework on which the basic functions operate and which is configured to record the basic function logs.

5. The image processing apparatus as claimed in claim 1, further comprising: a user interface unit configured to provide a user interface for making log recording settings that specify a combination of the basic functions, wherein the log recording unit is configured to record the basic function logs if the basic functions are used in the combination specified in the log recording settings.

6. The image processing apparatus as claimed in claim 1, further comprising: a user interface unit configured to provide a user interface for making log recording settings that specify the basic functions whose basic function logs are to be recorded, wherein the log recording unit is configured to record the basic function logs of the basic functions specified in the log recording settings.

7. The image processing apparatus as claimed in claim 1, further comprising: a user interface unit configured to provide a user interface for making log recording settings that specify one or more users, wherein the log recording unit is configured to record the application level function log and the basic function logs if the application level function is executed by any one of the users specified in the log recording settings.

8. The image processing apparatus as claimed in claim 1, further comprising: a user interface unit configured to provide a user interface for making log recording settings that define security levels, for each of which the basic functions whose basic function logs are to be recorded are specified, wherein the log recording unit is configured to determine whether to record the basic function logs according to the security levels defined in the log recording settings.

9. The image processing apparatus as claimed in claim 1, further comprising: a log displaying unit configured to display a list of image processing logs of image processing jobs performed by the application level function, a list of the basic function logs, or details of the image processing logs or the basic function logs.

10. A method for recording logs in an image processing apparatus where an application level function is implemented by one or more basic functions, the method comprising the step of: recording basic function logs of the basic functions according to log formats of the basic functions, the basic function logs being associated with an application level function log of the application level function.

11. The method as claimed in claim 10, further comprising the step of: attaching an image processing ID identifying an image processing job performed by the application level function to the application level function log and to each of the basic function logs, the basic function logs being associated with the application level function log by the image processing ID.

12. The method as claimed in claim 10, further comprising the step of: storing pointers to the basic function logs in the application level function log, the basic function logs being associated with the application level function log by the pointers.

13. The method as claimed in claim 10, wherein the basic function logs are recorded by a framework on which the basic functions operate, which framework is provided in the image processing apparatus.

14. The method as claimed in claim 10, wherein the basic function logs are recorded if the basic functions are used in a combination specified in log recording settings.

15. The method as claimed in claim 10, wherein the basic function logs are recorded if the basic functions are specified in log recording settings.

16. The method as claimed in claim 10, wherein the application level function log and the basic function logs are recorded if the application level function is executed by a user specified in log recording settings.

17. The method as claimed in claim 10, wherein whether to record the basic function logs of the basic functions is determined according to security levels defined in log recording settings, for each of which security levels the basic functions whose basic function logs are to be recorded are specified.

18. The method as claimed in claim 10, further comprising the step of: displaying a list of image processing logs of image processing jobs performed by the application level function, a list of the basic function logs, or details of the image processing logs or the basic function logs.

19. A storage medium having program code embodied therein for causing a computer controlling an image processing apparatus, in which image

processing apparatus an application level function is implemented by one or more basic functions, to record basic function logs of the basic functions according to log formats of the basic functions, the basic function logs being associated with an application level function log of the application level function.

Description:

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention generally relates to a technique of recording logs in an image processing apparatus such as a multifunction printer (MFP).

2. Description of the Related Art

Image processing apparatuses such as MFPs generally have a function to record various types of logs used, for example, to monitor usage of users or for accounting.

FIG. 1 shows exemplary software architecture of a conventional MFP. In the conventional MFP, software components are categorized into an application layer including applications representing tasks to be performed by users and a service layer including services for controlling hardware components. Each application in the application layer performs its task by using a combination of services in the service layer and sends a log including results of the services to a log manager to record the log. The number of applications or document input/output tasks (copying, printing, etc.) in the conventional MFP is limited to five to ten, and a log format is defined for each of the tasks.

FIG. 2 shows exemplary log formats in the conventional MFP. In FIG. 2, exemplary formats of the following logs are shown: a copy log, a scanning-storing log, a stored document printing log, a PC document printing log, a fax receiving-printing log, a fax transmission log, and a scanning-delivery log. Each format includes information indicating its log type: “copy” for the copy log, “scanning-storing” for the scanning-storing log, “stored document printing” for the stored document printing log, “PC document printing” for the PC document printing log, “fax receiving-printing” for the fax receiving-printing log, “fax transmission” for the fax transmission log, and “scanning-delivery” for the scanning-delivery log. Referring back to FIG. 1, a copy application records a copy log when copying a document; a printing application records a stored document printing log when printing a stored document or records a PC document printing log when printing a PC document; a fax application records a fax receiving-printing log when receiving a fax or records a fax transmission log when sending a fax; and a scanning application records a scanning-storing log when scanning and storing a document or records a scanning-delivery log when scanning and delivering a document.

Also, patent document 1 discloses another software architecture for recording and transferring logs.

[Patent document 1] Japanese Patent Application Publication No. 2006-107217

Meanwhile, an MFP employing software architecture different from that shown in FIG. 1 has been proposed. In the software architecture of the proposed MFP, software components are categorized into an activity group including activities that correspond to applications or application level functions in the conventional MFP described above; and a filter group including filters representing basic functions or functional units that are smaller than activities. The proposed MFP allows the user to compose an activity by combining filters and thereby to flexibly add application level functions.

FIG. 3 shows the software architecture of the proposed MFP in which each activity in the activity group is composed of a combination of filters in the filter group. For example, a copy activity is composed of a scanning filter and a printing filter.

Allowing users to add activities or application level functions may increase the security risks, and therefore it is all the more important to record logs in the proposed MFP.

One problem in recording logs in the proposed MFP is that since it allows users to compose new activities by combining filters, the number of necessary log formats increases and a program for recording logs becomes complicated if logs are recorded in the same manner as in the conventional MFP.

FIG. 4 shows exemplary log formats defined in the same manner as in the conventional MFP. Assuming that the proposed MFP includes four input filters (e.g. scanning, stored document input, PC document input, and fax reception) and four output filters (e.g. printing, email transmission, document storage, and fax transmission), the number of necessary log formats (or combinations of input and output filters) becomes $4 \times 4 = 16$.

The proposed MFP also enables outputting a document by multiple output filters (multiple output) and combining documents input from multiple input filters (multiple input). In this case, the number of filter combinations, i.e., the number of necessary log formats increases further.

FIG. 5 shows another exemplary log format defined in the same manner as in the conventional MFP. Assuming that up to four input filters and up