To be disregarded?

SOFTWARE SYSTEM MANAGEMENT DEVICE

BACKGROUND OF THE INVENTION

The present invention relates to software management device, and more particularly to a device for managing specifications of source programs composing software systems.

A software system is composed of a plurality of source programs. For example, a product delivery system may be composed of a source program for displaying a reception screen, a source program for processing reception process, and a source program for displaying a delivery screen. Source programs are produced separately, and then they are compiled and linked together to produce the software system.

A programmer produces a source program according to a document so called a program specification (hereinafter simply referred to as a specification). The specification includes a detailed statement of the data available for processing, the required outputs and so on. Examples of a program specifications are shown in "Encyclopedia of computer science and engineering, second edition", Van Nostrand Reinhold Company, New York, 1983, pp. 570 - 574.

The specifications are described in formats that are easy to understand such as a table. The specifications are saved even after the system is released and referred to for debugging or upgrading the software system.

Software manufacturers often produce a software system by slightly modifying a prior software system. They also release new versions of a software system to the same customer one after another. Thus, there are many variations of software systems of

the same type and, therefore, there are many variations of the specifications of the software system.

For identifying a software system from other software systems of the same type, a system name and a system revision number are given to each software system. For example, the system name identifies the client to which the software system was released. The system revision number (hereinafter referred to as system revision) identifies the version of a software systems that has the same system name.

The details of functions of a software system can be examined by looking through specifications of the software system. However, this task is apt to fall into difficulty because, once the system is released, it is often difficult to know the correspondence between software systems and specifications.

The difference of first and second software systems can be examined by comparing the specifications of the first and second software systems. However, this task is also apt to fall into difficulty because the volumes of the specifications are huge.

SUMMARY OF THE INVENTION

Therefore, one object of the present invention is to provide a software management device for examining the correspondence between software systems and specifications.

Another object of the present invention is to provide a software management device for examining the difference of two or more software systems.

According to the present invention, a software management device is connected to an input device, an output device and a

memory. The memory includes a configuration storage.

When the management device receives a data registration command from the input device, the management device registers data in the configuration storage of the memory according to information input from the input device.

The management device stores in the configuration storage a system identifier and a specification identifier corresponding to the system identifier when a source program identified by the specification identifier is included in a system identified by the system identifier.

When the management device receives a data reference command from the input device, the management device outputs specification information to the output device in response to the data reference command.

When "system configuration" function is chosen by the data reference command, the management device receives a system identifier from the input device. The management device reads from the configuration storage a specification identifier corresponding to the input system identifier. And then, the management device outputs the specification identifier to the output device.

When "system derivation list" function is chosen by the data reference command, the management device receives a system name from the input device. The management device reads from the configuration storage a system identifier including the input system name and outputs a system revision included in the system identifier to the output device.

When " specification derivation list" function is

implemented, the memory includes a specification storage for storing the specification identifier. When "specification derivation" function is chosen by the data reference command, the management device receives a specification name from the input device. The management device reads from the specification storage a specification identifier including the specification name and outputs a specification revision included in the specification identifier to the output device.

When "system search" function is chosen by the data reference command, the management device receives a specification identifier. The management device reads from the configuration storage a system identifier corresponding to the specification identifier and outputs the system identifier to the output device.

When "difference extraction" function is chosen by the data reference command, the management device receives first and second system identifiers of first and second software systems from the input device. The management device reads from the configuration storage a first group of specification identifiers corresponding to the first system identifier. The management device reads from the configuration storage a second group of specification identifiers corresponding to the second system identifier. The management device compares specification identifiers included in the first and second groups for determining a difference between the first and second groups. The management device outputs the difference between the first and second groups to the output device.

The management device outputs a specification identifier

included in both the first and second groups with a comment stating that the first and second software systems include identical source programs.

The management device outputs a specification identifier included only in the first group with a comment stating that a source program corresponding to the specification identifier is added to the first software system.

The management device outputs a first specification identifier in the first group when the second group includes a specification identifier including the same specification name as the first specification identifier but including different specification name. The first specification identifier is output with a comment stating that the first and second software system include different versions of a source program.

The memory may have a specification storage for storing the specification identifier and a content of a specification identified by the specification identifier. In this configuration, the management device reads from the specification storage a content of a specification and outputs the content to the output device.

BRIEF DESCRIPTION OF THE DRAWINGS

Other objects, features and advantages of the present invention will become more apparent when the following description is read in conjunction with the accompanying drawings, wherein:

FIG. 1 is a block diagram of a software management device according to a first embodiment of the present invention.

FIG. 2 shows an example of the contents of the configuration

specification information section 21.

FIG. 3 shows an example of the contents of the revision management information section 22.

FIG. 4 shows the structure of commands of the software management device.

FIG. 5 is a flowchart illustrating the processing of the data registration means 11.

FIG. 6 shows an example of a system configuration received by the data registration means 11.

FIG. 7 shows an example of the contents of the configuration storage 21 updated by the data registration means 11.

FIG. 8 shows an example of the contents of the specification storage 22 updated by the data registration means 11.

FIG. 9 is a flowchart illustrating an example of the processing of the data reference means 12.

FIG. 10 is an example of a menu displayed on the display device 3 when data is referenced.

FIG. 11 is a flowchart illustrating the processing of the search section 13.

FIG. 12 is an example of a menu displayed in the step 31 in fig. 11 on the display device 3.

FIG. 13 is a flowchart illustrating the processing of the search section 13 when the menu item "system configuration" is selected.

FIG. 14(a) and 14(b) show examples of screen images displayed in the steps 41 and 43 in fig. 13, respectively.

FIG. 15 is a flowchart illustrating the processing of the search section 13 when the menu item "system derivation list" is

selected.

FIG. 16(a) and 16(b) show examples of screen images displayed in the steps 51 and 53 in fig. 15, respectively.

FIG. 17 is a flowchart illustrating the processing of the search section 13 when the menu item "specification derivation list" is selected.

FIG. 18(a) and 18(b) show examples of screen images displayed in the steps 61 and 63 in fig. 17, respectively.

FIG. 19 is a flowchart illustrating the processing of the search section 13 when the menu item "system search" is selected.

FIG. 20(a) and (b) show examples of screen images displayed in the steps 71 and 73 in fig. 19, respectively.

FIG. 21 is a flowchart illustrating the processing of the difference extraction section 14.

FIG. 22 is a flowchart illustrating the processing of the difference extraction section 14.

FIG. 23 illustrates an example of the output screen output by the difference extraction section 14.

FIG. 24 is a flowchart illustrating the process for displaying the contents of a specification carried out by the difference extraction section 14.

FIG. 25 is a flowchart illustrating the processing of the second embodiment of the present invention.

FIG. 26 is a flowchart illustrating the detailed processing in the step 106 in fig. 25.

FIG. 27 (a) - 27 (c) illustrates an exemplary process of the second embodiment of the present invention.

FIG. 28 - 32 show examples of specifications.

In these drawings, the same reference numerals depict the same parts, respectively.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Referring to fig. 1, the software management device 1 according to the first embodiment of the present invention is a programmed digital computer. The management device 1 comprises data registration means 11 and data reference means 12. The data reference means 12 comprises search section 13 and the difference extraction section 14. The data registration means 11 and the data reference means 12 are modules of computer programs. The operations of the data registration means 11 and the data reference means 12 are described below.

The management device is connected to a display device 3, a keyboard 4, a pointing device 5, a floppy disk 6 and a storage device 2.

The storage device 2 has a configuration storage 21 and a specification storage 22.

Referring to fig. 2, the configuration storage 21 stores system configurations of released software systems.

In this specification, a system configuration of a software system means a combination of a system name and a system revision of the software system, and specification names and specification revisions of specifications of source programs composing the software systems.

As explained above, a combination of the system name and the system revision identifies a software system among many versions of the software system.

The specification name and the specification revision are

identifiers of a specification. A combination of the specification name and the specification revision identifies a specification among other specifications.

The specification names describe the function of the source program produced according to the specification. The specification revisions identify the version of the specification. In this embodiment, a specification corresponds to a source program on a one-to-one basis.

In the example in fig.2, the configuration storage 21 stores "Company A product delivery system" and "Company B product delivery system" as system names.

There are two versions of the company A product delivery system identified by the system revision "R1.1" and "R2.1", respectively. There are only one version of the company B product delivery system identified by the system revision "R1.1".

The company A product delivery system version R1.1 is composed of three source programs. The three source programs correspond to a reception screen specification version R2.1, a reception processing specification version R1.3, and a delivery screen specification version R3.1.2, respectively.

The company A product delivery system version R2.1 is composed of five source programs. The five source programs correspond to a reception screen specification version R3.1, a reception processing specification version R2.3, delivery screen specification version R3.1.2, a car allocation screen specification version R1.1, and a car allocation map specification version R1.1, respectively.

The company B product delivery system version R1.1 is

composed of four source programs. The five source programs correspond to a reception screen specification version R3.1, a reception processing specification version R1.4, a acceptance screen specification R1.1, and a delivery screen specification R3.1.2, respectively.

Referring to fig. 3, the specification storage 22 stores the specification names, specification revisions, and the contents of each specification.

In the example of fig. 3, "Reception screen specification", "Reception processing specification", "Delivery screen specification", "Car allocation screen specification", "Car allocation map screen specification" and "acceptance screen specification" are stored as system names in the specification storage 22.

The specification storage 22 stores three versions of the reception screen specification that are identified by the specification revisions of "R1.1", "R2.1" and "R3.1", respectively.

The specification storage 22 stores four versions of the reception processing specification that are identified by the specification revisions of "R1.1", "R1.2", "R1.3" and "R1.4", respectively.

The specification storage 22 stores four versions of the delivery screen specification that are identified by the specification revisions of "R1.1", "R2.1", "R3.1" and "R3.1.2".

The specification storage 22 stores the version "R1.1" of car allocation screen specification, car allocation map specification and acceptance screen specification.

Next, the operation of the management device 1 is described.

Referring to fig. 4, the management device 1 accepts a "data registration" command or a "data reference" command. When the operator inputs the data reference command, he chooses one of "search" and "difference extraction" functions. When the operator chooses "search", he chooses one of "system configuration", "system derivation list", "specification derivation list" and "system search" functions.

The data registration command is executed by the data registration means 11. The data reference command is executed by the data reference means 12. Especially, the data reference command is executed by the search section 13 of the data reference means 12 when "search" is chosen by the operator. The data reference command is executed by the difference extraction section 14 of the data reference means 12 when "data extraction" is chosen by the operator.

Next, the operation of the data registration means 11 is described.

When a system is released, the operator inputs the data registration command to update the contents of the configuration storage 21 and the specification storage 22. When the data registration command is input, the data registration means 11 requests the operator to input the system configuration of the newly released system and the contents of the specifications of the system. Receiving the system configuration and the contents of the specification via key board 4 or floppy disk drive 6, the data registration means 11 update the contents of the configuration storage 21 and the specification storage 22.

Next, more detailed operation of the data registration means 11 is described referring to fig. 5. In the steps 1 to 4 and 5 to 11, the data registration means 11 updates the configuration storage 21 and the specification storage 22, respectively.

Referring to fig. 5, in the first step, the data registration means 11 receive a system configuration and contents of specifications. As stated earlier, the system configuration of a software system includes a system name and a system revision of the system, and a specification names and specification revisions of the specifications of source programs composing the software system.

In the second step, the data registration means 11 determines whether the same system name as the received one is stored in the configuration storage 21 or not. When the same system name exists in the configuration storage 21, the data registration means 11 skips the step 3 and executes the step 4. When the same system name does not exist in the configuration storage 21, the data registration means 11 executes the step 3.

In the step 3, the data registration means 11 registers the system name in the configuration storage 21.

In the step 4, the data registration means 11 registers the received system revision, specification names and specification revisions into the configuration storage 21. The data registration means 11 establish the correspondence of the system name already registered with the system revisions, the specification names and the specification revisions registered.

In the step 5, the data registration means 11 selects one specification name among those received in the step 1.

In the step 6, the data registration means 11 determines whether all the specification name has already been selected and no more specification name can be selected in the step 5 or not. If so, the data registration means 11 terminates its operation. If not, the data registration means 11 executes the step 7.

In the step 7, the data registration means determines whether the same specification name as the selected one exists in the specification storage 22. If the selected specification name exists, the data registration means 11 executes the step 9. If the selected specification name does not exist, the data registration means 11 execute the step 8.

In the step 8, the data registration means 11 registers the selected specification name in the specification storage 22.

In the step 9, the data registration means 11 determines whether the specification revision registered in the specification storage 22 is the same as that of the selected specification name or not. When they are the same, the data registration means 11 executes the step 6 again. When they are different from each other, the data registration means 11 executes the step 10.

In the step 10, the data registration means 11 registers the specification revision of the selected specification name in the specification storage 22. The data registration means 11 establishes the correspondence of the specification name registered in the specification storage 22 with the specification revision.

In the step 11, the data registration means 11 registers the contents of the specification identified by the specification

name into the specification storage 22. The data registration means 11 establishes the correspondence of the specification name with the contents of the specification.

After executing the step 11, the data registration means 11 executes the step 6 again.

The steps 8 to 11 are executed with respect to all specification names in the received system configuration.

Next, an example of the operation of the data registration means 11 is described referring to fig. 6 to 8.

Assume the case that the data registration means 11 receive data shown in fig. 6 as a system configuration, and the configuration storage 21 and the specification storage 22 store the data shown in fig. 2 and 3, respectively.

In this case, the data registration means 11 skips step 3 because the received system name "Company B product delivery system" exists in the configuration storage 21 as depicted in fig. 2. Therefore, the data registration means 11 registers the system revision, the specification names and the specification revisions in the configuration storage 21 in the step 4. The contents of the configuration storage 21 is updated as shown in fig. 7 by this operation.

Referring to fig. 5 and 8, the step 10 is executed with respect only to the "acceptance screen specification" because the received specification revision and the registered specification revisions are the same with respect to other specification names. The data registration means 11 adds the specification revision "R2.1" and the contents of the acceptance screen specification version R2.1. The contents of the specification storage 22 is

updated as shown in the fig. 8 by this operation.

Next, the operation of the data reference means 12 is described.

Referring again to fig. 4, the operator inputs the data reference command when he refers to the data stored in the configuration storage 21 and the specification storage 22. Receiving the data reference command, the data management device 1 activates the data reference means 12.

Referring to fig. 9, in the step 21, the data reference means 12 displays a menu shown in fig. 10 on the display device 3. The operator selects one of the menu items by moving the pointing device 5.

In the step 22, the data reference means 12 determines whether one of the menu items is selected or not. When one of the menu items is selected, the data reference means 12 executes the step 23. When no menu item is selected, the data reference means 12 executes the step 24.

In the step 23, the data reference means 12 executes the processing corresponding to the selected menu item. If the "search" item is selected, the processing is executed by the search section 13. If the "difference extraction" item is selected, the processing is executed by the difference extraction section 14. After the execution, the data reference means 12 executes the step 21 again.

In the step 24, the data reference means 12 determines whether an exit instruction is input via the keyboard 4 by the operator or not. If the exit instruction is input, the data reference means 12 terminates its operation. If the exit

instruction is not input, the data reference means 12 executes the step 22 again.

Next, the operation of the data reference means 12 when the "search" menu item is chosen in the step 23 in fig. 9.

Referring to fig. 11, in the step 31, the search section 13 displays a menu as shown in fig. 12 on the display device 3.

The "system configuration" means the operation to display the specification names and the specification revisions of a specified software system.

The "system derivation list" means the operation to display all system revisions of a specified software system.

The "specification derivation list" means the operation to display all the revisions of a specification specified by a specification name.

The "system search" means the operation to display the system names and the system revisions of software systems including the source program produced according to a specification specified by the specification name and the specification revision.

In the steps 32 to 34, the search section 13 executes the same processing as the steps 22 to 24 in fig. 9, respectively.

Next is described the operation of the search section 13 when the "system configuration" item is selected in the step 33 in fig. 11.

Referring to fig. 13, in the step 41, the search section 13 displays an input screen shown in fig. 14(a) and requests the operator to input a system name.

In the step 42, the search section 13 receives a system name

and a system revision via keyboard 4 from the operator.

In the step 43, the search section 13 reads from the configuration storage 21 the specification names and the specification revisions included in the software system specified by the received system name and system revision. The search section 13 displays the specification names and the specification revisions on the display device 3.

In the step 44, the search section 13 determines whether a request to display the contents of the specifications is input by the operator or not. If the request is input, the search section 13 executes the step 45. If the request is not input, the search section 13 executes the step 46.

In the step 45, the search section 45 reads from the specification storage 22 the contents of the requested specification and displays it on the display device 6. After displaying the contents of the specification, the search section 13 executes the step 44 again.

In the step 46, the search section determines whether an exit instruction is input by the operator or not. If the exit instruction is input, the search section executes step 31 in fig. 11. If the exit instruction is not input, the search section executes the step 44 again.

Referring to fig. 14(a), when the search section 13 receives "company A product delivery system" and "R1.1" as the system name and the system revision, respectively, the search section 13 reads from the configuration storage 21 "reception screen specification R2.1", "reception processing specification R1.3" and "delivery screen specification R3.1.2" as the specification

names and the specification revisions.

Referring to fig. 14(b), the specification names and the specification revisions are displayed on the display device 3.

Next is described the operation of the search section 13 when the "system derivation list" item is selected in the step 33 in fig. 11.

Referring to fig. 15, in the step 51, the search section 13 displays an input screen shown in fig. 16(a) on the display device 3 and require the operator to input a system name.

In the step 52, the search section 13 receives the system name via the keyboard 4 from the operator.

In the step 53, the search section 13 read from the configuration storage 21 all system revisions of the received system name. The system revisions are displayed on the display device 3.

In the step 54, the search section 13 determines whether an exit instruction is input from the operator or not. If the exit instruction is input, the search section 13 executes the step 31 in fig. 11. If the exit instruction is not input, the search section 13 executes the step 54 again.

Referring to fig. 16(a), when the search section 13 receives "company A product delivery system" as the system name, the search section reads from the configuration storage 21, "R1.1" and "R2.1" as the system revision of the "company A product delivery system".

Referring to fig. 16(b), the search section 13 displays the system revisions on the display device 3.

Next is described the operation of the search section 13

when "the specification derivation list" is selected in the step 33 in fig. 11.

Referring to fig. 17, in the step 61, the search section 13 displays an input screen shown in fig. 18(a) on the display device 3 and request the operator to input a specification name.

In the step 62, the search section 13 receives a specification name via the keyboard 6 form the operator.

In the step 63, the search section 13 reads from the specification storage 22 all specification revisions of the received specification name. The specification revisions are displayed on the display device 3.

In the step 64, the search section 13 determines whether an exit instruction is input from the operator or not. If the exit instruction is input, the search section 13 executes the step 31 in fig. 11. If the exit instruction is not input, the search section 13 executes the step 64 again.

Referring to fig. 18(a), when the search section 13 receives "reception screen specification " as the specification name, the search section reads from the configuration storage 21 "R1.1", "R2.1" and "R3.1" as the specification revisions of the "reception screen specification".

Referring to fig. 18(b), the search section 13 displays the revisions on the display device 3.

Next is described the operation of the search section 13 when the "system search" item is selected in the step 33 in fig. 11.

Referring to fig. 19, in the step 71, the search section displays an input screen shown in fig. 20(a) and requests the

operator to input a specification name and a specification revision.

In the step 72, the search section 13 receives the specification name and the specification revision via the keyboard 6 from the operator.

In the step 73, the search section 13 selects one combination of a specification name and a specification revision in the configuration storage 21.

In the step 74, the search section 13 determines whether the step 75 to 76 is executed with respect to all specification names in configuration storage 21. If so, the search section 13 executes the step 77. If not, the search section executes the step 75.

In the step 75, the search section 13 determines whether the selected specification name and specification revision are the same as those received in the step 72 or not. If they are the same, the search section 13 executes the step 76. If they are not the same, the search section 13 executes the step 73 again.

In the step 76, the search section 13 reads from the configuration storage 21 the system name and the system revision corresponding to the selected specification name and the specification revision. The system name and the system revision are displayed on the display device 3. After executing the step 76, the search section 13 executes the step 73 again.

Referring to 20(a), when the search section 13 receives the "reception screen specification" and "R1.1" as the specification name and the specification revision, respectively, the search section 13 read from the configuration storage "company A product

delivery system" and "R1.1" as the system name and the system revision corresponding to the "reception screen specification" version "R1.1".

Referring to fig. 20(b), the search section 13 displays the system name and the system revision on the display device 3.

Next, the operation of the difference extraction section 14 is described. The difference extraction section 14 is activated when the "difference extraction" item is selected in the step 23 in fig. 9.

Referring to fig. 21, in the step 81, the difference extraction section 14 displays an initial state screen and requests the operator to input a first system name, a first system revision, a second system name and a second system revision. The initial state screen only includes "DIFFERENCE EXTRACTION", "FIRST SYSTEM" and "SECOND SYSTEM" of the display items in the screen shown in the fig. 23.

In the step 82, the difference extraction section 14 receives the first and second system names and the first and the second system revisions. If an exit instruction is input by the operator in the step 82, the difference extraction section 14 terminates its operation (the step 83). When the input of the first and second system names and revisions is completed, the difference extraction section 14 executes the step 84. Hereinafter, a system specified by the first system name and the first system revision is referred to as a first system. a system specified by the second system name and the second system revision is referred to as a second system.

In the step 84, the difference extraction section 14 reads

the specification names and the specification revisions of the first and second systems into storage A and B, respectively.

In the step 85, the difference extraction section 14 selects one set of a specification name and a specification revision (e.x., "reception screen specification, R3.1") from the storage A. The selected specification name and specification revision are removed from the storage A.

In the step 86, the difference extraction section 14 determines whether the selection in the step 85 is failed or not. If the selection is failed because there is no specification name remains in the storage A, the difference extraction section 14 executes the step 87. If the selection is succeeded, the difference extraction section 14 executes the step 92 in fig. 22.

Referring to fig. 22, in the step 92, the difference extraction section 14 determines whether the selected specification of the selected specification revision exists in the storage B. If exists, the difference extraction section 14 executes the step 93. If not exists, the difference extraction section 14 executes the step 95.

In the step 93, the difference extraction section 14 displays the selected specification name and revision with a comment "(identical)" in the output column of the first system.

In the step 94, the difference extraction section 14 removes the selected specification name and specification revision from the storage B. After the execution of the step 94, the difference execution section 14 executes the step 85 again.

In the step 95, the difference extraction section 14 determines whether the selected specification name exists in the

storage B. If exists, the difference specification section 14 executes the step 96. If not exists, the difference execution section 14 executes the step 98.

In the step 96, the difference specification section 14 displays the selected specification name and its specification revision in the storage A with a comment "(CHANGED)" in the output column of the first system. The difference extraction section 14 displays the specification revision of the selected specification name in the storage B in the output column of the second system.

In the step 97, the difference extraction section 14 removes the selected specification name and its specification revision from the storage B. After the execution of the step 97, the difference extraction section 14 executes the step 85 in fig. 21.

In the step 98, the difference extraction section 14 displays the selected specification name and the selected specification revision with a comment "(ADDED)" in the output column of the first system. After the execution of the step 98, the difference execution section 14 executes the step 85 in fig. 21.

Referring again to fig. 21, in the step 87, the difference extraction section 14 determines whether any specification name exists in the storage B or not. If exists, the difference extraction section 14 executes the step 88. If not exists, the difference extraction section 14 executes the step 89.

In the step 88, the difference extraction section 14 displays all specification names and their specification revisions in the output column of the second system.

In the step 89, the difference extraction section 14 determines whether the operator inputs a request to display the contents of the specification or not. If the operator inputs, the difference extraction section 14 executes the step 90. If the operator does not requests, the difference extraction section 14 executes the step 91.

In the step 90, the difference extraction section 14 displays the contents of the specification specified by the operator. The operator specifies one of the specification displayed on the display device 3 by using the pointing device 5. The detailed operation of the difference extraction section 14 in the step 90 is described below. After the execution of the step 90, the difference extraction section 14 executes the step 89 again.

In the step 91, the difference extraction section 14 determines whether an exit instruction is input by the operator or not. If the exit instruction is input, the difference extraction section 14 terminates its operation and the data reference means 12 executes the step 21 in fig. 9.

Next is described an example of the operation of the difference extraction section 14.

If, for example, the contents of the configuration storage 21 are as shown in FIG. 2 and if the "company A product delivery system, R2.1" and the "company B product delivery system, R1.1" are input as the first and second system names and revisions, "reception screen specification, R3.1", "reception processing specification, R2.3", "delivery screen specification, R3.1.2", "car allocation screen specification, R1.1" and "car allocation

map specification, R1.1" are stored in the storage A, while "reception screen specification, R3.1", "reception processing specification, R1.4," "acceptance screen specification, R1.1" and "delivery screen specification, R3.1.2" are stored in the storage B.

In the first stage, the "reception screen specification R3.1" is selected in the step 85. In this stage, "reception screen specification R3.1 (IDENTICAL)" is displayed as shown in fig. 23 in the step 93 because the same specification name of the same specification revision exists in the storage B. The "reception screen specification R3.1" is removed from the storage A and B in the step 85 and 94, respectively.

In the second stage, the "reception processing specification R2.3" is selected in the step 85. In this stage, "reception processing specification R2.3 (CHANGED)" is displayed as shown in fig. 23 in the step 96 because the same specification name of a different specification revision "reception processing specification R1.4" exists in the storage B. Simultaneously, "R1.4" is displayed in the output column of the second system.

The "reception processing specification R2.3" and "reception screen specification R1.4" is removed from the storage A and B in the steps 85 and 97, respectively.

In the third stage, the "delivery screen specification R3.1.2" is selected in the step 85. In this stage, "delivery screen specification R3.1.2 (IDENTICAL)" is displayed as shown in fig.23 in the step 93 because the same specification name of the same specification revision exists in the storage B. The "delivery screen specification R3.1.2" is removed from the

storage A and B in the steps 85 and 94, respectively.

In the fourth stage, the "car allocation screen specification R1.1" is selected in the step 85. In this stage, "car allocation screen specification R1.1 (ADDED)" is displayed as shown in fig. 23 in the step 98 because the specification name "car allocation screen specification" does not exists in the storage B. The "car allocation screen specification R1.1" is removed form the storage A in the step 98.

In the fifth stage, the "car allocation map specification R1.1" is selected in the step 85. In this stage, "car allocation map specification R1.1 (ADDED)" is displayed as shown in fig. 23 in the step 98 because the same specification name does not exists in the storage B. The "car allocation map specification R1.1" is removed from the storage A in the step 98.

In the sixth stage, the difference extraction section 14 executes the step 88 because no specification name exists in the storage A and "acceptance screen specification R1.1" exists in the storage B. In the step 88, "acceptance screen specification R1.1" is displayed in the output column of the second system as shown in fig. 23.

Next is described the detailed operation of the difference extraction section 14 in the step 90.

Referring to fig. 24, in the step 101, the difference extraction section 14 determines whether the specified specification is in the output column of the first system. If the specification is in the output column of the first system, the difference extraction section 14 executes the step 102. If the specified specification is in the output column of the second

system, the difference extraction section 14 executes the step 105.

In the step 102, the difference extraction section 14 reads the contents of the specified specification from the specification storage 22 and displays it in the output column of the first system.

In the step 103, the difference extraction section 14 determines whether the specified specification is modified version of a specification of the second system or not. That is, the difference extraction section 14 determines whether the comment "(CHANGED)" is displayed on the right of the specified specification name and specification revision. If the comment "(CHANGED)" is displayed, the difference extraction section 14 executes the step 104. If the comment "(CHANGED)" is not displayed, the difference extraction section 14 terminates the operation of the step 90.

In the step 104, the difference extraction section 14 reads from the specification storage 22 the contents of specification in the storage B with the same specification name as that of the specified specification of the first system. The difference extraction section 14 displays the contents of the specification in the output column of the second system.

In the step 105, the difference extraction section 14 reads the contents of the specified specification form the specification storage 22 and displays it in the output column of the second system. After the execution of the step 105, the difference extraction section 14 ends its operation of the step 90.

Next, the advantages of the first embodiment is described.

First, the operator can examine the correspondence between software systems and specifications due to the search section 13, the configuration storage 21 and the specification storage 22. This search facility makes it possible for the operator to identify the specifications composing a specific software system, and to identify software systems including a specific specification. The search facility also makes it possible to display all the revisions of a specific system name or specification.

Second, the operator can examine the contents of the specification soon after the search processing because the specification storage 22 stores the contents of the specification.

Third, the specification storage 22 does not take up large space in the storage device 2. This advantage is due to the division of the specification into the units corresponding to source programs. When a source program is modified, only the specification corresponding to the modified source program is added in the specification storage 22.

Fourth, the operator can easily identify the difference between two software system. This facility is due to the specification name and the specification revision given to each specification, and to the difference extraction section 14.

Next, modifications of the first embodiment is described.

First, the contents of the specification can be removed from the specification storage 22.

Second it is possible to modify the displayed image shown

in fig. 10, 12, 14(a), 14(b), 16(a), 16(b), 18(a), 18(b), 20(a), 20(b) and 23.

Third, the first embodiment is preferably applied to a system in which a source program is automatically produced from a specification. In this kind of system, it is easy to store the contents of the specification in the specification storage 22 because the specifications are not drafted on paper but coded as computer readable format.

Next, the second embodiment of the present invention is described.

Referring to fig. 25, the feature of the second embodiment resides in the step 106 executed after the step 104 in fig. 24. In the step 106, the difference between the specifications of the first and second systems is identified. The detailed operation of the difference extraction section 14 in the step 106 is described below.

Referring to fig. 26, in the step 111, the difference extraction section reads one item of the contents of the specification of the first system.

In the step 112, the difference extraction section 14 reads one item of the contents of the specification of the second system.

In the step 113, the difference extraction section 14 determine whether the items read in the steps 111 and 112 are the same or not. If they are the same, the difference extraction section 14 executes the step 115. If they are not the same, the difference extraction section 14 executes the step 114.

In the step 114, the difference extraction section 14

displays the item in the second system in reverse video.

In the step 115, the difference extraction section 14 determines whether the step 111 to 114 is done with respect to all items of the contents of the specification or not. If the steps are executed with respect to all items, the step 106 terminates. If the steps are not executed with respect to all items, the difference extraction section 14 executes the step 111 again.

Referring to fig. 27(a) and 27(b), when the contents of the specifications of the first and second systems are as shown in fig. 27(a) and 27(b), respectively, the "data file" items are different from each other. Therefore, the difference extraction section displays the "data file" item of the second system in reverse video as shown in fig. 27(c).

According to the second embodiment, the operator can easily examine the difference of two systems.

The second embodiment is preferably applied to a system in which a source program is produced automatically from a specification. In this kind of system, the comparison of the items is easy because programmers draft specifications in a standard format.

Next, examples of the specifications applicable to the present invention are described.

Referring to fig. 28 showing an example of a program specification. The program specification describes the operation of a program. The program specification may include a data flow diagram.

Referring to fig. 29 showing an example of an action

definition specification, the action definition specification defines the operation of each action. An "action" is a command executed in a system in which a source program is produced from a specification.

Referring to fig. 30 showing another example of the program specification, this program specification describes the correspondence between items in a file and items in an output datum.

Referring to fig. 31 showing an example of a file specification, the file specification describes the contents and formats of records in files.

Referring to fig. 32 showing an example of a screen specification, the screen specification describes the format of a displayed image.

The present invention can be applied to other kinds of specifications not shown in figures such as a data base specification, a structure definition specification, a procedure specification, an input/output specification, a format specification, a data specification.

The database specification describes the format of data stored in a database. The structure definition specification describes the structure of a software system. The procedure specification describes the operation of each procedure. The input/output specification describes the format and protocol of the input/output data. The format specification describes a format printed on a printer. The data specification describes the format and structure of data.

The present embodiments are therefore, to be considered in

all respects as illustrative and not restrictive, the scope of the invention being indicated by the appended claims rather than by the foregoing description and all changes which come within the meanings and range of equivalency of the claims are therefore intended to the embraced therein.



What is claimed is:

1. A software managing method implemented in a software management device, said software management device being connected to an input device, an output device and a memory, said memory including a configuration storage, said method comprising the steps of:

receiving a data registration command from said input device and registering data in said configuration storage of said memory according to information input from the input device; and

storing in said configuration storage a system identifier and a specification identifier corresponding to said system identifier when a source program identified by said specification identifier is included in a system identified by said system identifier.

2. A software managing method implemented in a software management device, said software management device being connected to an input device, an output device and a memory, said memory including a configuration storage, said method comprising the steps of:

storing in said configuration storage a system identifier and a specification identifier corresponding to said system identifier, said specification identifier corresponding to said system identifier when a source program identified by said specification identifier is included in a system identified by said system identifier;

receiving a data reference command from said input device; and

outputting specification information to said output device

* C L M S *

in response to said data reference command.

3. A method according to claim 2, said step for outputting comprises the steps of:

receiving a system identifier from said input device;

reading from said configuration storage a specification identifier corresponding to said system identifier; and

outputting said specification identifier to said output device.

4. A method according to claim 2 wherein said system identifier includes a system name and a system revision, said step for outputting comprises the steps of:

receiving a system name from said input device;

reading from said configuration storage a system identifier including said system name; and

outputting a system revision included in said system identifier to said output device.

5. A method according to claim 2, wherein said memory has a specification storage for storing said specification identifier, said specification identifier includes a specification name and a specification revision, said step for outputting comprises the steps of:

receiving a specification name from said input device;

reading from said specification storage a specification identifier including said specification name; and

outputting a specification revision included in said specification identifier to said output device.

6. A method according to claim 2, said step for outputting comprising steps of:

* C L M S *

receiving a specification identifier;

reading from said configuration storage a system identifier corresponding to said specification identifier; and

outputting said system identifier to said output device.

7. A method according to claim 2, said step for outputting comprising the steps of:

receiving first and second system identifiers of first and second software systems from said input device;

reading from said configuration storage a first group of specification identifiers corresponding to said first system identifier;

reading from said configuration storage a second group of specification identifiers corresponding to said second system identifier; and

comparing specification identifiers included in said first and second groups for determining a difference between said first and second groups; and

outputting said difference between said first and second groups to said output device.

8. A method according to claim 7, wherein said step for outputting comprises the step of:

outputting to said output device a specification identifier included in both said first and second groups.

9. A method according to claim 8, wherein said step for outputting a specification identifier comprises the step of:

outputting to said output device a comment stating that said first and second software systems include identical source programs.

10. A method according to claim 7, wherein said step for outputting comprises the step of:

outputting to said output device a specification identifier included only in said first group.

11. A method according to claim 10, wherein said step for outputting a specification identifier comprises the step of:

outputting to said output device a comment stating that a source program corresponding to said specification identifier is added to said first software system.

12. A method according to claim 7, wherein said step for outputting comprises the step of:

outputting to said output device a first specification identifier in said first group when said second group includes a specification identifier including the same specification name as said first specification identifier but including different specification name.

13. A method according to claim 12, wherein said step for outputting a first specification identifier comprises the step of:

outputting to said output device a comment stating that said first and second software system include different versions of a source program.

14. A method according to claim 2, wherein said memory has a specification storage for storing said specification identifier and a content of a specification identified by said specification identifier, said step for outputting comprises a steps of:

reading from said specification storage a content of a specification; and

outputting said content to said output device.

15. A software management device comprising:

a storage device having a configuration storage and a specification storage, said configuration storage storing, for software systems of the same type, a system name and all of the existing system revisions for each client to which the system has been released and also storing names and revisions of the specifications corresponding on a one-to-one basis to the source programs constituting each system revision, said specification storage storing a specification storage for recording a name and all of the existing specification revisions of each specification;

an input device;

an output device;

a data registration means for registering data in said configuration storage and said specification storage according to input information from said input device; and

a data reference means form outputting a content of said configuration storage and said specification storage according to data reference requests from said input device.

16. A software management device according to claim 1, wherein said data reference means references a name and revision of the specification constituting the first software system identified by the system name and revision specified in a data reference request from said input device and the second software system identified by the system name and revision specified in the request to output the difference in the name and revision of the specifications constituting the first and second software

systems.

17. A software management device comprising:

a storage device having a configuration storage and a specification storage, said configuration storage storing, for software systems of the same type, a system name and all of the existing system revisions for each client to which the system has been released and also storing names and revisions of the specifications corresponding on a one-to-one basis to the source programs constituting each system revision, said specification storage storing a name, all of the existing specification revisions, and contents of each specification;

an input device;

an output device;

a data registration means for registering data in said configuration storage and said specification storage according to input information from said input device; and

a data reference means for outputting a content of said configuration storage and said specification storage according to data reference requests from said input device.

18. A software management device according to claim 1, wherein said data reference means reference said name and revision of the specification constituting the first software system identified by the system name and revision specified in a data reference request from said input device and the second software system identified by the system name and revision specified in the request to output the difference in said name, revision, and contents of the specifications constituting the first and second software systems.

* C L M S *

19. A software managing method implemented in a software management device, said software management device being connected to an input device, an output device and a memory, said memory including a configuration storage and a specification storage, said method comprising the steps of:

receiving a data registration command from said input device for registering a software system;

storing in said configuration storage a system identifier and at least one specification identifier corresponding to said system identifier input from said input device; and

storing in said specification storage a content of a specification identified by said specification identifier, the content of the specification stored in said specification storage being linked to the specification identifier stored in said configuration storage.

20. The software managing method recited in claim 19 wherein the step of receiving a data registration command further includes the step of prompting the user to input a system configuration of the software system and the content of the specification.

21. The software managing method recited in claim 20 wherein the step of prompting the user includes prompting for input of a system revision identifier and prompting for input of a specification revision identifier, and wherein the step of storing in said configuration storage includes storing the system revision with said system identifier and storing the specification revision with said specification identifier.

22. The software managing method recited in claim 21 further comprising the steps of:

* C L M S *

determining whether a same system identifier as a user input system identifier has been previously registered in said configuration storage; and

registering the user input system identifier by storing the user input system identifier in said configuration storage if not previously registered in said configuration storage.

23. The software managing method recited in claim 22 further comprising the steps of:

determining whether a same specification identifier as a user input specification identifier has been previously registered in said configuration storage; and

registering the user input specification identifier by storing the user input specification identifier in said configuration storage if not previously registered in said configuration storage.

24. The software managing method recited in claim 23 further comprising the steps of:

determining whether a same specification revision as a user input specification revision has been previously registered in said configuration storage; and

registering the user input specification by storing the user input content of the specification in said specification storage if not previously registered in said configuration storage.



25. A software managing method substantially as hereinbefore described with reference to the accompanying drawings.

26. A software management device substantially as hereinbefore described with reference to the accompanying drawings.