9800323-9
POOR COPY

METHOD AND SYSTEM FOR SOFTWARE DEVELOPMENT AND SOFTWARE DESIGN EVALUATION SERVER

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to a method and system for software development and a software design evaluation server therefor. More particularly, the present invention relates to a method and system for software development which allow users to verify the functional specifications during the development process, and also to a software design evaluation server which facilitates user participation in a software development process to verify the functional specifications of an application under development.

2. Description of the Related Art

In the context of business process reengineering (BPR) including analysis and redesign of business rules and practices, many organizations recognize the necessity of reconfiguration and redeployment of their business computing systems to adapt to the new business processes. Information system departments in such organizations are requested to put their efforts to improvement of productivity in the software development processes, while the end users are also encouraged to develop a system by themselves according to their own needs.

In reality, however, actual user needs are so

diversified that information system departments are overwhelmed by ever-increasing development work backlogs. The delivery of maximum efficiency in software development is therefore an urgent requirement to be achieved.

With respect to the software development methodologies, a conventional "waterfall" process has been popularly used to develop business application programs, in which each development phase strictly follows its predecessor. That is, a software development process starts with repeated discussions between the user and developer so as to clarify the user's requirements and finalize the product target specifications, and then proceeds to the next phases such as design, coding, and test in a stepwise manner.

Today, prototyping has been widely introduced, on the other hand, as an integral part of a software development process. In this methodology, the users are requested to evaluate a prototype of an application under development and provide their comments and suggestions on the features presently implemented therein, thus allowing the developer to modify and improve the program in a timely manner according to the user's feedback. The prototyping methodology promotes heightened user involvement in the product definition and development processes, while ensuring satisfaction with the final application.

The above-described software development with

prototyping phases requires users to participate in the system development from its early stage to achieve a high quality level in the final product. This user participation, however, entails movement of human resources and/or instruments from one place to another, thus imposing some burdens on both users and developers.

Another problem is availability of users. They are not always ready for frequent participation in the system development. Insufficient user participation, however, reduces the opportunities for in-process check and correction, which may result in some indefinite functional specifications remaining unclarified or necessary features left missing in the final product.

SUMMARY OF THE INVENTION

Taking the above into consideration, a first object of the present invention is to provide a convenient software design evaluation server which ensures that all functions requested by the user be fully implemented in the application under development.

To accomplish the first object, according to the present invention, there is provided a software design evaluation server, coupled to a client computer via a network, which allows a user to make a review of functional specifications of an application during a development process thereof. This software design evaluation server comprises: (a) comment-space inclusive

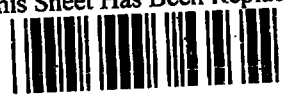
THIS SHEET HAS BEEN REPLACED

prototype presentation means for generating a comment-space inclusive prototype by adding a comment space to a prototype of the application under development to allow the user to write comments and suggestions about features of a screen or data items included in the screen as part of the prototype, and for sending the comment-space inclusive prototype to the client computer for presentation to the user; (b) identification data storage means for storing information for identifying the screen and the items, as well as information for identifying development engineers involved in the development of the application; and (c) user feedback distribution means for receiving from the user the comments and suggestions about the screens and/or the items from the user through the network, and distributing the received comments and suggestions to the development engineers, based on the information stored in said identification data storage means.

A second object of the present invention is to provide a convenient software development method which assures that all functions requested by the user will be implemented in the application under development.

To accomplish the second object, according to the present invention, there is provided a software development method which allows a user to make a review of functional specifications of an application during a development process thereof. This software development

This Sheet Has Been Replaced



THIS SHEET HAS BEEN REPLACED

method comprises the steps of: (a) presenting a prototype of the application under development to the user over a network; (b) sending the user's comments and suggestions about the prototype over the network; (c) correcting the application under development so that the user's comments and suggestions be reflected therein, whereby the development process advances toward a final version of the application; and (d) delivering the final version of the application to the user via the network.

A third object of the present invention is to provide a convenient software development system which assures that all functions requested by the user will be implemented in the application under development.

To accomplish the third object, according to the present invention, there is provided a software development system which allows a user to make a review of functional specifications of an application during a development process thereof.

This software development system comprises: (a) a client computer comprising (a1) software development order means for placing an order for software development via a network, (a2) prototype display means for displaying a prototype of the application under development on a monitor screen, and (a3) user feedback notification means for sending comments and suggestions concerning functional specifications for the prototype over the network; and (b) a server comprising (b1) similar case searching means for

This Sheet Has Been Replaced



THIS SHEET HAS BEEN REPLACED

accepting the software development order from the user and retrieving a similar case that relates to the ordered software development, (b2) process content determination means for determining contents of the application, based on the similar case that has been retrieved by the similar case searching means, (b3) quotation means for issuing a price quotation of the software development, (b4) authentication key assignment means for giving an authentication key to the user, when the user has accepted the price quotation and a software development contract has been made, (b5) prototype presentation means for presenting the prototype to the user via the network, and (b6) user feedback distribution means for distributing the comments and suggestions received from the client computer to development engineers in charge of the software development.

The above and other objects, features and advantages of the present invention will become apparent from the following description when taken in conjunction with the accompanying drawings which illustrate preferred embodiments of the present invention by way of example.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a diagram showing the principle of a software design evaluation server of the present invention;

FIG. 2 is a flowchart showing an operational

This Sheet Has Been Replaced

procedure of a first embodiment;

FIG. 3 is a flowchart showing an operational procedure of a second embodiment;

FIG. 4 is a flowchart showing an operational procedure of a third embodiment;

FIG. 5 is a diagram showing an example of a screen presented by a comment-space inclusive prototype presentation unit;

FIG. 6 is a flowchart showing a process where the user's comments and suggestions about prototype specifications are notified as user feedback and reflected in the prototype;

FIG. 7 is a diagram showing how a user feedback table is associated with a development team management table;

FIG. 8 is a diagram showing how the user feedback is distributed to development engineers;

FIG. 9 is a flowchart showing how the user feedback is reflected in a prototype in the third embodiment;

FIG. 10 is a flowchart showing a process executed by a software development method according to the present invention;

FIG. 11 is a diagram showing a concept of a software development system according to the present invention;

FIG. 12 is a flowchart showing a process executed

by the software development system;

FIG. 13 is a diagram showing a similar case table;

FIG. 14 is a diagram showing another similar case table;

FIG. 15 is a diagram showing still another similar case table;

FIG. 16 is a diagram showing a data item table;

FIG. 17 is a diagram showing a software specification table;

FIG. 18 is a diagram showing another software specification table;

FIG. 19 is a diagram showing still another software specification table;

FIG. 20 is a diagram showing how a price quotation is processed by a quotation unit;

FIG. 21 is a diagram showing how a prototype presentation unit reports the development progress status and presents a prototype to the user;

FIG. 22 is a diagram showing an initial menu of a personnel information management system;

FIG. 23 is a diagram showing a search condition entry screen through which a first kind of search conditions are entered to the personnel information management system;

FIG. 24 is a diagram showing another search condition entry screen through which a second kind of search conditions are entered to the personnel information

management system;

FIG. 25 is a diagram showing still another search condition entry screen through which a third kind of search conditions are entered to the personnel information management system;

FIG. 26 is a diagram showing a summary of matches screen; and

FIG. 27 is a diagram showing a personnel database screen.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

Several embodiments of the present invention will be described below with reference to the accompanying drawings.

FIG. 1 explains the concept of a software design evaluation server of the present invention. More specifically, FIG. 1 illustrates a simple client/server model comprising a software design evaluation server 10, a client computer 20 at a user's site, and a network 30 for interconnecting them. It is assumed here that a personnel management application is now in development and its prototype 11a is available for the purpose of evaluation of functional specifications. The software design evaluation server 10 supports prototype evaluation by the user during the process of software development.

The software design evaluation server 10 comprises a comment-space inclusive prototype presentation unit 11

which generates a special form of prototype by adding a comment space 11b to the original prototype 11a to allow the user to put his/her comments and suggestions about the proposed screens design or the items included in the prototype screens. This prototype is referred to as the comment-space inclusive prototype. The comment-space inclusive prototype presentation unit 11 sends the comment-space inclusive prototype to the client computer 20 via the network 30 for prototype review by the user.

The software design evaluation server 10 further comprises an identification data storage unit 12 which stores information to identify various screen designs under development and items included in each screen and to identify development engineers involved in the project.

Serving as another element of the software design evaluation server 10, a user feedback distribution unit 13 receives comments and suggestions about the screens or items from the user through the network 30, associates them with the identification data stored in the identification data storage unit 12, and distributes them as user feedback to the development engineers involved in the project.

In the present invention outlined above, the term "prototype" is defined as a software program (or application) at any development stage. Such software development starts with selection of an appropriate template system, or standard version of business

applications, which is considered to be most suitable for the user's needs. The developer further implements the user's specific requests into this template system, as well as receiving the user's comments and suggestions during the development.

A first embodiment of the present invention proposes how to interact with the user to conduct a review of functional specifications for an application under development. FIG. 2 is a flowchart showing the procedure of this first embodiment.

[S1] The comment-space inclusive prototype presentation unit 11 transmits a comment-space inclusive prototype to the client computer 20.

[S2] The comment-space inclusive prototype is downloaded and installed into the client computer 20.

[S3] The prototype screens are presented to the user, along with a space for comments and suggestions. The user reviews the features implemented in the prototype screens and writes his/her comments and suggestions in the given space. The client computer 20 returns the user's comments and suggestions as user feedback information to the software design evaluation server 10.A second embodiment of the present invention proposes another method of interacting with the user to conduct a review of the functional specifications. FIG. 3 is a flowchart showing the procedure of this second embodiment.

[S10] The comment-space inclusive prototype

presentation unit 11 provides the user with a comment-space inclusive prototype by using an Internet browser with Java language capabilities. (Java is a trademark of Sun Microsystems, Inc.)

[S11] The user reviews the prototype and writes his/her comments and suggestions in the given space. This user feedback information is sent from the client computer 20 to the software design evaluation server 10.

A third embodiment of the present invention proposes still another method of interacting with the user to conduct a review of the functional specifications. FIG. 4 is a flowchart showing the procedure of this third embodiment.

[S20] The comment-space inclusive prototype presentation unit 11 first establishes a communication link between the user's client computer 20 and another client computer of one developer who is in charge of some screens or items under development. It then makes a replica of a prototype screen now displayed on the developer's terminal and sends it to the client computer 20 for user review.

[S21] The user's comments and suggestions about the prototype screen are returned to the user feedback distribution unit 13 through a chat messaging service over the network 30, and the user feedback distribution unit 13 directly forwards them to the relevant development engineer. That is, the user can send his/her comments and

suggestions to the relevant development engineer in a realtime manner.

The next few paragraphs will describe a typical screen presented by the comment-space inclusive prototype presentation unit 11 in the first and second embodiments. FIG. 5 shows an example screen 20a presented to the user for review of the functional specifications.

This example screen 20a includes two windows; one shows a prototype 11a and the other shows a comment space 11b prepared for the user to write his/her comments and suggestions on the prototype 11a. The prototype 11a shown in FIG. 5 is a personal information screen as part of an application under development, entitled the "Personnel Information Management System." The prototype 11a comprises several items such as an employee's name, address, face picture, etc.

On the other hand, the comment space 11b has a subject line to indicate the screen name "Personal Information" and the corresponding screen ID "0006." Being arranged in the form of a table, the comment space 11b has three columns for item IDs, check boxes, and comments.

The item IDs listed in the first column are the identification numbers of data items shown in the prototype 11a. For example, the face picture has a code of "001" and the employee's name has a code of "002." The check boxes in the second column are used by the user to

indicate whether each item is OK or NG (No Good). The space in the third column allows the user to enter his/her comments and suggestions about the prototype 11a. Suppose that, for example, the user prefers a full-length picture to the face picture as part of the screen design. The user expresses his/her preference by marking "NG" for the item ID 001 and writing his/her instruction into the comment space, such as "Use a full-length photograph."

Once the user feedback on the functional specifications are provided, they must then be reflected in the application design. The following description will explain how this is done specifically in the first and second embodiments of the present invention. FIG. 6 is a flowchart showing this process.

[S30] The comments and suggestions given by the user are delivered from the client computer 20 to the software design evaluation server 10.

[S31] The user feedback distribution unit 13 checks the user ID and password.

[S32] With the user ID and the password, the user feedback distribution unit 13 confirms that the sender of the comments is the person who is in charge of prototype review. After this user authentication, it examines the user ID, reference number of the user feedback, screen ID, and item ID, in order to locate particular development engineers relating to the screen or items that were subjected to the prototype review. The

user feedback distribution unit 13 then distributes the user's comments and suggestions to the engineers.

The identification data storage unit 12 stores a user feedback table 12a and a development team management table 12b. FIG. 7 shows how these tables are associated with each other.

The user feedback table 12a stores comments and suggestions collected from the user, each of whose entries consists of a reference number assigned to each comment, screen ID, item ID, and the body of each comment. Note that the user feedback table 12a is created separately for individual user IDs. The one shown in FIG. 7 is of the user ID "USER01."

The development team management table 12b stores information on who is in charge of each screen design. Each entry of the table 12b consists of a reference number, screen ID, and developer ID.

The user feedback table 12a and development team management table 12b are associated with each other by the common values of the reference number and screen ID. For instance, the user feedback table 12a shows that a user having a user ID "USER01" has provided a comment on the screen ID0006 of the reference number 00001, and the development team management table 12b indicates that this screen is developed by a development engineer having a developer ID "B0002."

FIG. 8 is a diagram showing how the user's

comments and suggestions are distributed to the relevant development engineers. The user feedback table 12a in FIG. 7 is divided into four blocks 12a-1 to 12a-4 in FIG. 8, according to the developer IDs stored in the development team management table 12b. Those divided tables are distributed to four different development engineers as identified by their developer IDs. The user's comment explained in FIG. 5, for example, will be delivered to the development engineer with a developer ID B0002, since the comment is directed to the item ID 001 in the screen ID 0006. This allows the development engineer to make a modification of the screen design, so that a full-length picture will be used instead of the present face picture.

Next, the following description will explain how the user's comments and suggestions on the functional specifications are reflected in the design of the prototype 11a in the third embodiment. FIG. 9 is a flowchart showing this process.

[S40] The comment-space inclusive prototype presentation unit 11 directly displays the prototype 11a on the client computer 20 by replicating what is developed on the development engineer's computer. More specifically, the personal computer at the development engineer's site is directly connected to another personal computer at the user's site over the network 30, just as in the desktop conferencing (DTC) or personal computer-based videoconferencing. The information such as user ID,

reference number, and screen ID is used as parameters for this direct connection. The relevant developers can be located based on this information in the same way as described in FIGS. 7 and 8.

[S41] The user directly communicates his/her comments and suggestions to the development engineers by using a chat messaging service over the network 30, which enables what the user types to appear on the developer's computer screen in a realtime manner.

As described above, the software design evaluation server 10 of the present invention offers a prototype to the user on an online basis, along with a space for comments and suggestions, thereby permitting the user to return feedback via the network 30. The software design evaluation server 10 distributes this user feedback information on the prototype specification to appropriate development engineers to notify them of the problems that have to be solved. This structural arrangement allows software users to participate in prototype evaluation as part of a software development process in a more convenient way, and thus makes it possible to prevent any defects in functional specifications from happening to the application under development.

Next, a software development method of the present invention will be explained below. FIG. 10 is a flowchart showing a process executed by a software development method according to the present invention, which comprises

the following steps.

[S50] A prototype software program is presented online to the user via the Internet or other networks.

[S51] The user evaluates the prototype and sends his/her comments and suggestions online.

[S52] The prototype is corrected so that the notified comments and suggestions be reflected therein. The software development advances toward the final version by repeating the above processes.

[S53] The completed software product is delivered to the user over the network and subjected to the acceptance procedure by the user.

In the software development method of the present invention, all interactions with the user are performed on an online basis, including prototype presentation, collection of user's comments and suggestions, and delivery of final products as explained above. This structural arrangement allows users to participate in prototype evaluation as part of a software development process in a more convenient way, and thus makes it possible to prevent any defects in functional specifications from happening to the application under development.

Next, a software development system of the present invention will be explained below. FIG. 11 shows the concept of a software development system of the present invention. This software development system takes a

client/server network structure involving a client computer 40 and a server 50 to provide an improved software development environment where the developers cooperate with the users in reviewing the functional specifications of a target application in the process of software development.

The client computer 40 comprises a software development order unit 41, a prototype display unit 42, and a user feedback notification unit 43. The software development order unit 41 places an order for software development through the network 30. The prototype display unit 42 displays a prototype of the application under development on a monitor screen. The user feedback notification unit 43 accepts the user's comments and suggestions concerning the functional specifications for the prototype and sends them over the network 30.

The server 50 comprises, on the other hand, the following functional elements: a similar case searching unit 51, a process content determination unit 52, a quotation unit 53, an authentication key assignment unit 54, a prototype presentation unit 55, and a user feedback distribution unit 56. The similar case searching unit 51 accepts a software development order from the user and retrieves similar case data that may be applicable to the ordered software development. The process content determination unit 52 determines the content of a process, such as user interface screens, based on the similar case

data that has been retrieved. The quotation unit 53 automatically issues a price quotation for the software development that was ordered. When the user has accepted the price quotation and the developer has made a software development contract with the user, the authentication key assignment unit 54 then submits an authentication key to the user. The prototype presentation unit 55 presents a prototype to the user over the network 30. The user feedback distribution unit 56 distributes the user's comments and suggestions collected from the client computer 40 to the development engineers in charge of the software development.

The operation of the above software development system is represented in a flowchart of FIG. 12.

[S60] The software development order unit 41 places an order for software development through the network 30. What is referred to as the network 30 is the Internet.

[S61] Upon receipt of the software development order from the user, the similar case searching unit 51 searches for similar case data. The process advances to step S62 if any similar case data is found. Otherwise, the process goes to step S63.

[S62] The process content determination unit 52 determines the contents of user interface screens etc., based on the similar case data.

[S63] The process content determination unit 52

newly defines the contents of the application process.

[S64] The quotation unit 53 estimates the software development cost, possible delivery schedule, and the like, and presents the quotation to the user through the network 30.

[S65] When the user has accepted the quotation and the developer has made a software development contract with the user, the authentication key assignment unit 54 gives an authentication key to the user.

[S66] The prototype presentation unit 55 sends a prototype to the user via the network 30.

[S67] The prototype display unit 42 displays the prototype on a monitor screen.

[S68] The user feedback notification unit 43 accepts the user's comments and suggestions concerning the functional specifications for the prototype and sends them through the network 30.

[S69] The user feedback distribution unit 56 distributes the user's comments and suggestions collected through the client computer 40 to the development engineers in charge of the software development.

The following few paragraphs will be devoted to the similar case searching unit 51. This similar case searching unit 51 makes a keyword search to find similar case data that matches with the application to be developed for the user. The usable keywords may include various industry classifications (e.g., manufacturing,

distribution, financial, etc.), job categories (e.g., general affairs, sales, personnel affairs, etc.), names of application systems (e.g., business management, sales management, personnel management, etc.), and names of software instruments (e.g., Access, Notes, Java, etc.). (note: MS Access is a trademark of Microsoft Corporation. Lotus Notes is a trademark of Lotus Development Corporation. Java is a trademark of Sun Microsystems, Inc.)

FIGS. 13 to 15 are diagrams showing similar case tables 51a to 51c. As shown in FIGS. 13 to 15, the similar case tables 51a to 51c are structured in a normalized relational model, and stored collectively in a similar case database.

The similar case searching unit 51 retrieves similar case data out of these similar case tables 51a to 51c. Assume here that a database search is to be conducted with keywords specified as: "manufacturing" for industry classification, "personnel affairs" for job category, "Access" for software instrument, and "personnel management" for application system name. Then a search operation will proceed in the following way.

The similar case table 51a of FIG. 13 consists of reference numbers, industry classifications, job categories, application system names, and software instruments. The first entry with the reference number "00001" matches with the present search condition.

The similar case table 51b of FIG. 14 consists of reference numbers, screen IDs, and screen names, whose entries are arranged in accordance with the reference number. Take the entries with the reference number "00001" for instance. They describe that a screen with ID number "0005" is entitled "Summary Information" and that another screen with ID number "0006" is titled "Individual Information."

The similar case table 51c of FIG. 15 consists of reference numbers, screen IDs, item IDs, item names, and properties, whose entries are arranged in accordance with the screen ID. The property columns have different definitions depending on the nature of each item. For example, when an item of interest is a character string, the properties will describe whether it is expressed in Japanese or English, or in what color it should be presented. More specifically, in FIG. 15, the item "Face Picture" has a "Monochrome" property, while the item "Name" has "Japanese" and "Gothic Font" properties.

The similar case searching unit 51 searches for similar case data to deeper levels by using a data item table in the following way. FIG. 16 shows a data item table 51d, as part of a data item database, which is structured in a normalized relational model. This data item table 51d is composed of detailed data items extracted from the above-mentioned similar case database, with classification according to the names of application

systems and screens that they belong to.

More specifically, each entry of the data item table 51d comprises a reference number, application system name, data item ID, data item name, properties, and implementation number. It should be noted here that this table may contain a plurality of similar entries having the same application system name. Such similar entries have different data item names or unequal attribute values depending on their implementations.

For example, the first and second entries have the same application system name "Personnel Management" and the same data item name "Name." These two similar entries, however, differ from each other in the values of property #1, or the number of characters, because their definitions depend on their respective implementations, "00020" and "00001."

For another example, the identification number given to each member of the staff is called "Employee Number" in one implementation "00020." On the other hand, the same number is called "Personnel ID" in another implementation "00001". As such, the same kind of entries may have different data item names.

As clarified above, the data item table 51d collects many detailed data items extracted from the similar case database, thus allowing the similar case searching unit 51 to determine the data items by comparing them in more detailed similarity levels. If no perfect

match is obtained, the similar case searching unit 51 creates a new item and registers it to the data item database.

Next, the process content determination unit 52 will be described in detail. The process content determination unit 52 determines what the application in development is expected to do, based on the similar case retrieved by the similar case searching unit 51. The determined details of the process are then stored in the software specification database. FIGS. 17 to 19 show the software specification tables 52a to 52c as part of a software specification database that serves as the main storage for the information obtained from the users.

The software specification table 52a of FIG. 17 contains reference numbers and base implementation numbers. Here, the reference numbers are the order numbers received from users, while the base implementation numbers are the numbers of the similar case data which will be used as the basis to develop the ordered application. FIG. 17 specifically shows that a specific application with a reference number (or order number) "000001" will be developed with reference to two similar implementations "00001" and "00020" as the design basis.

FIG. 18 shows another software specification table 52b, which consists of base implementation numbers, screen IDs, item IDs, and comments, being arranged in accordance with the reference numbers. As seen in FIG. 18, this table

52b is a collection of user's comments and suggestions. Take the third entry with the base implementation number "00001" for example. It describes a change in the software specification that the item "001" should be a full-length picture.

FIG. 19 shows still another software specification table 52c, which is used for the entire system administration. This software specification table 52c contains a reference number, application system name, software instruments, databases used, and the number of users. The software specification table 52c of FIG. 19 specifically shows a project with a reference number "000001" which is titled "Personal Information Management System." This system is developed with "Access" software and makes access to "Oracle" database system, while ten users are expected to use the application. (note: Oracle is a trademark of Oracle Corporation.)

Referring next to FIG. 20, the quotation unit 53 will be explained below. FIG. 20 illustrates an entire client/server system, where the quotation unit 53 disposed in the server 50 estimates the development cost upon inquiry from the client computer 40.

At the user's site, a quotation parameter table 40b is displayed on a screen 40a of the client computer 40, allowing the user to enter the values for cost estimation. FIG. 20 shows that the user has given a first value *X* for the number of screens, a second value *Y* for the number of

reports or forms, and "Access" for software instrument, to describe a new application that the user is planning to order. Based on these key parameters, the server 50 estimates the cost required for the application development.

The quotation unit 53, as part of the server 50, comprises quotation calculation unit 53a and quotation delivery unit 53b. For example, the quotation calculation unit 53a estimates the development cost by using a formula as

$$Pe = (Psc \times X + Pre \times Y) \times Co \quad \ldots\ldots(1)$$

where *Pe* is the estimated development cost, *Psc* is the unit price per screen, *X* is the number of screens, *Pre* is the unit price per report or form, *Y* is the number of reports or forms, and *Co* is a coefficient that represents complexity of the process. Note that the unit price parameters *Psc* and *Pre* are dependent on what kind of software instrument is specified.

The quotation delivery unit 53b responds to the user's inquiry by supplying a price quotation calculated according to the above formula (1). The price quotation is delivered to the user in an immediate response mode via the Internet or in a delayed response mode via e-mail.

Next, the prototype presentation unit 55 will be explained below. The prototype presentation unit 55 presents a prototype to the user on an online basis, as well as informing him/her of the development progress

status.

FIG. 21 is a diagram showing how the prototype presentation unit 55 reports the development progress and presents a prototype to the user. First, the prototype presentation unit 55 displays a progress check dialog 55a on a first screen 42a, which dialog has text input boxes prompting the user to enter his/her user ID and password. When the user enters them and he/she is authenticated as the right person that he/she claims to be, the scene is changed to a second screen 42b.

The screen 42b shows the latest development status 55b. In FIG. 21, the latest development status 55b reports the following situations: the menu screen has been completed on February 3; the summary screen is 50 percent finished as of February 5 but still under way; and work for the personal information screen has not yet started as of February 5.

The display proceeds to a third screen 42c when the user selects the item "Menu Screen" from among those listed in the latest development status 55b. This screen 42c shows the "Menu Screen" 55c which is actually titled "Personnel Information System" along with some remarks such as "Complete and tested" given by a developer.

Referring back to the latest development status 55b, the user may select the item "Summary Screen." Then the display is changed to a fourth screen 42d, where a "Personnel Management Summary" 55d is presented along with

some remarks such as "Event handler not finished" given by another developer.

Referring next to FIGS. 22 to 27, several user interface screens will be explained below as an example application developed with the software development system of the present invention. More specifically, FIGS. 22 to 27 show the screens provided in a personnel information management system.

FIG. 22 is a diagram showing an initial menu of this personnel information management system. This initial menu 100 comprises a SEARCH button 101, an EXIT button 102, and a password input box 103. The SEARCH button 101 initiates a search operation. The EXIT button 102 terminates the present session and allows the user to exit from the personnel information management system. The password input box 103 is a text field to which the user enters his/her password.

FIG. 23 shows a first search condition entry screen 110 where a first kind of search conditions will be entered to the personnel information management system. More specifically, the first kind of search conditions include employee name, years of employment, managerial positions, and functional classes. In the first search condition entry screen 110, a name field 111 is used to specify the name of an employee to be searched for. The user can designate a temporal search range by entering specific years to a year of employment field 112. In this

field 112, the user is also allowed to select either "Any Years" or "Specific Years" option by clicking the corresponding check box. When he/she selected the "Any Years" option, the server 50 searches the records of the company's founding year through the present date. If the "Specific Years" option is selected, the server 50 only searches the records of the specified year(s). A managerial position field 113 is used to specify the managerial positions of employees to be searched for. Likewise, a class field 114 is used to designate the ratings of employees to be searched for.

A LOCATION button 115 is used to set additional search conditions about the office/factory locations where the targeted employees may be stationed. An ASSIGNMENT button 116 also allows additional search conditions to be defined, concerning the organizational units which they may belong to. These LOCATION and ASSIGNMENT buttons 115 and 116 will actually call up separate screens 120 (see FIG. 24) and 130 (see FIG. 25) which are designed for setting the extended search conditions as will be described later. A RESET button 117 clears all entry data in this screen. An EXIT button 118 allows the user to leave the search condition entry screen 110. A SEARCH button 119 starts the search operation as defined in this first search condition entry screen 110.

FIG. 24 shows a second search condition entry screen 120 where a second kind of search conditions will

be entered to the personnel information management system. This second search condition entry screen 120 lists a plurality of offices and factories 121 where the targeted employees may be stationed. The user can include those locations into the search conditions by clicking each item.

OK button 122 allows the user to return to the previous search condition set screen 110 of FIG. 23 after specifying the locations in the second search condition entry screen 120. SELECT ALL button 123 selects all of the places 121, while RESET button 124 clears all the present selections in turn.

FIG. 25 shows a third search condition entry screen 130 where a third kind of search conditions will be entered to the personnel information management system. This third search condition entry screen 130 lists a plurality of divisions and departments. A headquarters field 131 contains five check boxes to enable a search of individual departments in the headquarters. Likewise, a sales division field 132 contains five check boxes to enable a search of individual sales-related departments in the sales division. The first product development division field 133 contains four check boxes to enable a search of individual departments in the first product development division. The second product development division field 134 contains four check boxes to enable the search of individual departments in the second product development division.

OK button 135 allows the user to return to the previous search condition set screen 110 of FIG. 23 after specifying the locations in the third search condition entry screen 130. SELECT ALL button 136 is used to select all departments included in the headquarters field 131, sales division field 132, first product development division field 133, and second product development division field 134. RESET button 137 clears all the present selections.

FIG. 26 is a diagram showing a summary of matches screen 140. A summary of matches 141 shows individual ID codes, names, offices, assigned sections, managerial positions, and employment dates. PREVIOUS PAGE button 142 leads the user back to the previous page of the summary of matches 141. In turn, NEXT PAGE button 143 brings the user to the next page of the summary of matches 141. DETAILS button 144 calls up the detailed contents of the row of one individual code.

FIG. 27 is a diagram showing a personnel database screen 150. This screen 150 provides the user with the fully detailed information about a specific employee searched and selected by the user. NEXT PAGE button 151 and PREVIOUS PAGE button 152 allow him/her to move to the next page or previous page of the personnel database, respectively. PRINT button 153 is used to print out a personnel data table 150a that is presently displayed on the screen 150. SUMMARY button 154 takes the user back to

the summary of matches 141.

As described above, the software development system of the present invention receives an online order for application development. It then searches for similar cases and issues a quotation to the user on an online basis. During the software development process, the system provides the user with some prototypes for evaluation purposes, and the user's comments and suggestions are fed back to the application under development. This structural arrangement allows the user to participate in the software development process in a more convenient way, and thus makes it possible to prevent any defects in functional specifications from remaining in the final version of the application.

The above discussion will be summarized as follows. According to the present invention, the software design evaluation server offers online a prototype to the user, along with a space for comments and suggestions, thereby allowing the user to return his/her feedback over the network. The software design evaluation server distributes the user's comments and suggestions on the prototype specification to relevant development engineers to notify them of the problems and new requests that have to be solved and implemented. This structural arrangement of the server allows the user to participate in the software development process in a more convenient way, and thus makes it possible to prevent any defects in functional

specifications from remaining in the final version of the application.

Further, in the software development method of the present invention, all interactions with the user are performed on an online basis, including prototype presentation, collection of user's comments and suggestions, and delivery of final products. This structural arrangement allows the user to participate in the software development process in a more convenient way, and thus makes it possible to prevent any defects in functional specifications from remaining in the final version of the application.

Moreover, the software development system of the present invention receives an order for application development, searches for similar cases, and issues a quotation to the user, completely on an online basis. During the software development process, the system provides the user with some prototypes for evaluation purposes, and the user's comments and suggestions are fed back to the application under development. This structural arrangement of the system allows the user to participate in the software development process in a more convenient way, and thus makes it possible to prevent any defects in functional specifications from remaining in the final version of the application.

The foregoing is considered as illustrative only of the principles of the present invention. Further, since

numerous modifications and changes will readily occur to those skilled in the art, it is not desired to limit the invention to the exact construction and applications shown and described, and accordingly, all suitable modifications and equivalents may be regarded as falling within the scope of the invention in the appended claims and their equivalents.

* C L M S *

THIS SHEET HAS BEEN REPLACED

WHAT IS CLAIMED IS:

1. A software design evaluation server, coupled to a client computer via a network, which allows a user to make a review of functional specifications of an application during a development process thereof, comprising:

comment-space inclusive prototype presentation means for generating a comment-space inclusive prototype by adding a comment space to a prototype of the application under development to allow the user to write comments and suggestions about features of a screen or data items included in the screen as part of the prototype, and for sending the comment-space inclusive prototype to the client computer for presentation to the user;

identification data storage means for storing information for identifying the screen and the items, as well as information for identifying development engineers involved in the development of the application; and

user feedback distribution means for receiving from the user the comments and suggestions about the screens and/or the items from the user through the network, and distributing the received comments and suggestions to the development engineers, based on the information stored in said identification data storage means.

2. The software design evaluation server

This Sheet Has Been Replaced



according to claim 1, wherein said user feedback distribution means is activated when the user has successfully presented a correct user ID and a correct password that were previously given to the user.

3. The software design evaluation server according to claim 1, wherein said comment-space inclusive prototype presentation means sends the comment-space inclusive prototype to the client computer, makes the comment-space inclusive prototype be installed into the client computer, and presents the comment-space inclusive prototype to the user after the installation is finished.

4. The software design evaluation server according to claim 1, wherein said comment-space inclusive prototype presentation means sends the comment-space inclusive prototype to the client computer to present the comment-space inclusive prototype to the user by using an Internet browser.

5. The software design evaluation server according to claim 4, wherein said comment-space inclusive prototype presentation means sends to the client computer the comment-space inclusive prototype created by using a browser program with Java language capabilities.

6. The software design evaluation server

THIS SHEET HAS BEEN REPLACED

according to claim 1, wherein said comment-space inclusive prototype presentation means establishes a communication link between the client computer of the user and another client computer of the development engineer, based on the information stored in said identification data storage means, and presents the same screen as displayed on the second-mentioned client computer of the development engineer on the first-mentioned client computer of the user.

7. The software design evaluation server according to claim 6, wherein said user feedback distribution means directly distributes said comments and suggestions provided by the user, to the development engineer by using a messaging service.

8. A software development method which allows a user to make a review of functional specifications of an application during a development process thereof, comprising the steps of:

(a) presenting a prototype of the application under development to the user over a network;

(b) sending the user's comments and suggestions about the prototype over the network;

(c) correcting the application under development so that the user's comments and suggestions be reflected therein, whereby the development process

This Sheet Has Been Replaced

THIS SHEET HAS BEEN REPLACED

advances toward a final version of the application; and

(d) delivering the final version of the application to the user via the network.

9. A software development system which allows a user to make a review of functional specifications of an application during a development process thereof, comprising:

(a) a client computer comprising

software development order means for placing an order for software development via a network;

prototype display means for displaying a prototype of the application under development on a monitor screen, and

user feedback notification means for sending comments and suggestions concerning functional specifications for the prototype over the network; and

(b) a server comprising

similar case searching means for accepting the software development order from the user and retrieving similar case data that relates to the ordered software development,

process content determination means for determining contents of the application, based on the similar case data that is retrieved by said similar case searching means,

quotation means for issuing a price quotation of

This Sheet Has Been Replaced

* C L M S *

the software development,

authentication key assignment means for giving an authentication key to the user, when the user has accepted the price quotation and a software development contract has been made,

prototype presentation means for presenting the prototype to the user via the network, and

user feedback distribution means for distributing the comments and suggestions received from the client computer to development engineers in charge of the software development.

10. The software development system according to claim 9, wherein said prototype presentation means provides the user with development progress status of the software development in addition to the prototype.

THIS SHEET HAS BEEN REPLACED

This Sheet Has Been Replaced