Content Distribution System

CROSS REFERENCE TO RELATED APPLICATIONS

All the contents disclosed in Japanese Patent Application No. 2000-98106 (filed on March 31, 2000), including specification, claims, drawings and abstract and summary are incorporated herein by reference in its entirety.

FIELD OF THE INVENTION

This invention relates to a content distribution server, which distributes content to terminal devices.

BACKGROUND OF THEINVENTION

Fig. 1 shows a conventional content distribution system. In the system, each of terminals 2,4,6,8,and 10 in different type may access to each of corresponding servers 14, 16, 18, 20, and 22 via the Internet 12.

The personal computer (hereinafter referred to as PC) 2 which uses a browser for access, an i-mode (trademark) terminal 4, a WAP (trademark) terminal 6, a Personal Handy phone System (hereinafter referred to as PHS) terminal 8, and a terminal 10 for personal computer communications service need to access respectively to the server 14 for PC web, the server 16 for i-mode, the server 18 for WAP, the server 20 for PHS, and the server 22 for personal computer communications service. i-mode is mobile internet access services in Japan, by which contents on the Internet can be viewed using a browser equipped in its mobile phones as terminals. When a terminal device access to an inappropriate server, there is a possibility that either of no display or improper display is made. This is because display restrictions of each terminal device are different from one another.

Thus, users of such terminals need to change one of the severs to be accessed depending on the type of the terminal even when all the users try to access to the identical content. With the conventional system, another problem, such that desired content can not view properly when the content is prepared solely for a terminal(s) having different display restrictions, is also observed.

Suppliers of content, in contrast, need to prepare contents written for all kinds of terminals and they have to provide servers for storing these contents in order to make the contents to be viewed by many users. It is, therefore, creation of the contents, maintenance, and updating thereof require a lot of work to the suppliers.

It is an object of the present invention to provide a server for distributing content, by which suitable content for each of terminals is distributed without providing servers for each of the terminals. The aim of the present invention is, to solve the problems described above, and to provide a language converting system, which facilitate creation of content and update thereof.

SUMMARY OF THE INVENTION

(1) In accordance with characteristics of the present invention, there is provided a content server receiving an access by different types of terminal devices comprising:

content storage means, storing content;

terminal identifying means, identifying a display restriction of the terminal device accessing to the server; and

data format adjustment and transmission means, adjusting a data format of the content read out from the content storage means so as to match a display restriction of the terminal device depending on the

display restriction of the terminal device and transmitting the adjusted content to the terminal device.

In this way, proper display can be performed at each of the display terminals by just preparing and recording in the content server common contents independent from the display restrictions of the terminal devices.

(4) In accordance with characteristics of the present invention, there is provided a content server in which the display restriction is a size of content which can be stored temporarily at the terminal device for its display,

and wherein the data format adjustment and transmission means divides the content in accordance with the size of the content.

Thus, proper display can be performed by each of the terminal devices by just preparing common contents regardless of the display restrictions of the terminal devices.

(5) In accordance with characteristics of the present invention, there is provided a content server in which the data format adjustment and transmission means generates relevant information between the divided content when the content is divided, and wherein the relevant information is sent to the terminal device.

Consequently, the divided content can be viewed efficiently at the terminal device.

(6) In accordance with characteristics of the present invention, there is provided a content server in which the terminal identifying means detects the display restriction of the terminal by identifying a

type of the terminal device.

The display restriction can be detected by just identifying the type of the terminal device.

(7) In accordance with characteristics of the present invention, there is provided a content server in which the terminal identifying means identifies the type of the terminal device in accordance with a header of Hyper Text Transfer Protocol (HTTP) sent from the terminal.

Thus, the type of the terminal device can accurately be identified because the HTTP header indicates the protocol used by the terminal device.

(8) In accordance with characteristics of the present invention, there is provided a content server in which the terminal identifying means identifies the type of the terminal device in accordance with an IP address sent from the terminal device.

Type of the terminal devices can be identified accurately because the way of assigning IP addresses is unique among communication common carriers that define the type of the terminal devices.

(12) In accordance with characteristics of the present invention, there is provided a data format adjusting device for adjusting a data format of content, wherein the device is designed to receive a display restriction of a terminal to which the content is transmitted and to adjust the data format of the content read out from content storage means so as to match the display restriction of the terminal device depending on the display restriction thereof.

In this way, proper display can be performed at each of the display terminals by just preparing and recording in the content server common contents independent from the display restrictions of the terminal devices.

(15) In accordance with characteristics of the present invention, there is provided a data format adjusting device in which the display restriction is a size of content which can be stored temporarily at the terminal device for its display, and wherein the adjustment is performed by dividing the content in accordance with its size.

Thus, proper display can be performed by each of the terminal devices by just preparing common contents regardless of the display restrictions of the terminal devices.

(16) In accordance with characteristics of the present invention, there is provided a data format adjusting device wherein the device generates relevant information between the divided content when the content is divided.

Consequently, the divided content can be viewed efficiently at the terminal device.

(17) In accordance with characteristics of the present invention, there is provided a method of distributing content to different types of terminal devices, the method comprising the steps of:

storing the content;

detecting an output restriction of the terminal device accessing thereto;

adjusting a data format of the content so as to match the output restriction of the terminal device depending on the output restriction of

the terminal, and

transmitting the adjusted content to the terminal device.

In this way, proper display can be performed at each of the display terminals by just preparing and recording in the content server common contents independent from the display restrictions of the terminal devices.

The term "adjustment of a data format for content" in this invention refers mainly to an adjustment performed to adjust a data format of content. It includes a concept, division of content, change in the number of characters on a line in content and similar changes.

The term "terminal identifying means" refers to means for obtaining information on output restrictions of terminal device. It also forms a concept including means for indirectly obtaining output restrictions of a terminal device as a result of retrieving the type of the terminal device and/or means for receiving information on the output restrictions transmitted by the terminal device.

The term "output restrictions" refers to restrictions imposed on the output of the content. It also forms a concept including restrictions on display that of audio output, that of printing, writing of data and so on.

The term "type of terminal device" refers to a type of processing performed on display languages and so on of contents. It also forms a concept including difference of processing in not only hardware but also that of software. In addition, such difference in type is observed even between terminals manufactured by the same manufacturer and in operation under the same protocol because the capacity thereof for temporarily use is differ from others due to difference in the version of its

hardware and/or software.

Other objects and features of the present invention will be more apparent to those skilled in the art on consideration of the accompanying drawings and following specification, in which are disclosed several exemplary embodiments of the present invention. It should be understood that variations, modifications and elimination of parts might be made therein as fall within the scope of the appended claims without departing from the spirit of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a diagram of a conventional content distribution system;

Fig. 2a is a diagram showing an example of a content distribution system using a content server according to the present invention;

Fig. 2b is a block diagram showing an example of the content distribution server according to the present invention;

Fig. 3 is a diagram showing the overall structure of basic language data;

Fig. 4 is a diagram showing the structure of the basic language data;

Fig. 5 is a diagram showing the structure of tags in the basic language data;

Fig. 6 is a diagram showing an example of the body of the basic language;

Fig. 7 is a diagram showing an example of a style sheet of the basic language;

Fig. 8 is a diagram showing an example of the body of the basic language;

Fig. 9 is a flowchart of the processes performed by a content distribution server for distributing content;

Fig. 10 is a flowchart of a program for identifying the type of

terminals;

Fig. 11a is a flowchart of a program for generating information on session and for controlling transition of web pages;

Fig. 11b is another flowchart of a program for generating information on session and for controlling transition of web pages;

Fig. 11c is a flow chart of a program for performing division of content;

Fig. 11d is a view showing the limitations in size at each terminal;

Fig. 12 is a view showing how to identify a session ID;

Fig. 13 is a view showing content thus converted in its format;

Fig. 14 is an example of session information;

Fig. 15 is an image displayed on a PHS terminal device;

Fig. 16 is a view illustrating a result of conversion in format;

Fig. 17 is a view showing an example of information on sessions;

Fig. 18 is another image displayed on the display of a PHS terminal device;

Figs.19a,19b are views showing content written in the basic language;

Figs. 20a, 20b are views showing converted content;

Fig. 21 is a view showing session information;

Fig. 22 is a view showing edited session information;

Figs.23a, 23b are views showing session information being generated as a result of performing division of content;

Fig. 24 is a view showing a divided content; and

Fig. 25 is a flowchart of a program for converting a language into another language;

Fig. 26 is a diagram showing the structure of the basic language data;

Fig. 27 is a view showing the tree structure of the basic language data;

Fig. 28 is a view showing the tree structure of the basic language

data;

Fig. 29 is a view showing data converted into the tree structure;

Fig. 30 is a view showing data converted into the tree structure;

Fig. 31 is a flowchart of a program for converting basic language data into HTML (Hyper Text Markup Language) data;

Fig. 32 is a view showing the converted HTML data;

Fig. 33 is a diagram showing an image displayed on the screen with the HTML data shown in Fig. 26;

Figs. 34a, 34b are flowcharts of a program for converting basic language data into HDML (Handheld Device Markup Language) data;

Fig. 35 is a view showing the HDML data thus converted;

Fig. 36 is a diagram showing an image displayed on the screen with the HDML data depicted in Fig. 29;

Fig. 37 is a flowchart for performing division of content;

Figs. 38a, 38b are views showing converted content;

Fig. 39 is a view showing divided content;

Fig. 40 is a view showing another divided content;

Fig. 41 is a view showing the content distribution system according to the present invention in another embodiment;

Fig. 42 is a view showing the content distribution system according to the present invention in another embodiment.

DETAILED DESCRIPTION OF THE INVENTION

1. Content distribution system

An example of a content distribution system according to the present invention is depicted in Fig. 2a. Connected to the Internet 12 is a server 30 for distributing content (hereinafter referred to as content server). Also, terminal devices 2, 4, 6, 8, and 10 each different in type are connectable to the Internet 12.

The personal computer (hereinafter referred to as PC) 2 is

accessible to the content server 30 via a protocol converting device 40 in an electric-mail format (that is, in both SMTP/POP3). The PC 2 is also accessible to the content server 30 using a browser software (that is, in http protocol). Further, mobile phones 4, 6, and 8 are respectively accessible to the content server 30 via protocol converting devices 42 and 44 (see Fig. 2b) in the forms of "i-mode cellular phone system", "WAP", and electric mail". The PC 10 is accessible to the content server 30 via a protocol converting device 46.

The content server 30 comprises terminal identifying means 33, language conversion means 35, content storage means 37 and data format adjusting and transmission means 41. The terminal identifying means 33 identifies the type of terminals accessing. The language conversion means 35 converts content written in the basic language into one written in a display language used by a terminal device depending on the type of the terminal device thus identified. The data format adjusting and transmission means 41 adjusts on a real time basis the converted content which is read out from the content storage means 37 and then converted into a certain display language so as to match its display restrictions with the ones of the accessing terminal device depending on the type thereof. In contrast, content which does not require conversion in language (such as a file consist solely of text data), content which is read out from the content storage means 37 is adjusted on a real time basis. Subsequently, the data format adjusting and transmission means 41 transmits the adjusted content to the terminal device.

In this embodiment, content stored in the server is written in the basic language not depending on the type of the terminal device, and the language is converted into other language when the content is transmitted to the terminal device. In this way, content written in a

specific language can be viewed on all the terminals different in type.

Further, content is adjusted in response to the display restriction of the terminal device when it is transmitted to the terminal device. Thus, it is necessary for the creator of contents to consider the display restriction of the terminal devices when they create the contents.

Fig. 2b is a diagram showing the overall system and a block diagram of the hardware for the content server 30. The content server 30 comprises a communication adapter 32 for connecting to the Internet 12, a CPU 34, a memory 36, and a hard disk 38. Stored in the hard disk 38 are the operating system (such as UNIX etc.), a program for identifying terminals, a program for converting language, a program for adjusting data format and so on. In addition, content written in the basic language is stored therein. The terminal type identification program performs a process in which type of the terminal accessing is identified by collaborating with the operating system. Also, the language conversion program carries out a process in which the basic language data is converted into the data written in a language appropriate to the display language used with each of the terminals in association with the operating system. The data format adjusting program is a program in association with the operating system for performing processing in which data format of content is adjusted so as to match with display restrictions of each terminal device.

2. Structure of basic language data

(1) Overall stricture

In this embodiment, data of the content stored in the hard disk 38 is written in the basic language. Fig. 3 shows the structure of such data. The basic language data comprises a style sheet (display format description data) describing a display format thereof and the body

representing the contents of displays (display contents description data). The style sheet is a part describing color for display, size of characters, display formats and so on. The body is a part representing a text describing its contents. The style sheet, however, is provided to the data optionally. It is meant that the data will be displayed in the standard format when none of the style sheet is provided.

The use of such basic language ease the generation of management information. This also facilitates a processing which converts content written in the basic language into that written in a display language for a terminal device. The management information, however, may also be generated in accordance with content written in the basic language previously stored into the content storage means 37.

In this embodiment, the body of the data is common regardless of the display language after conversion in language, but the style sheet is unique to the display language after the conversion. Thus, it is necessary to provide a style sheet for data each written in HTML and HDML respectively when the display format of these data other than the standard format is desired.

Hence, when a style sheet just for the HTML data is provided therefor, the standard display format is employed on the HDML data during conversion.

In other embodiment, a common style sheet may be used regardless of the display language used after conversion.

(2) Body of data

The descriptive structure of a body is shown in Fig. 4. The body comprises tags (parts surrounded by "<" and ">") and texts (parts other

than the tags). The tags, as shown in Fig. 5, start with "<", then the name of the tags (English characters; essential) and both attributes follow thereafter, and end with ">". The attributes includes these, the name of the attributes (English characters), "=", and attribute values (arbitrary characters). Attributes are optionally provided to the data, the data can do without attributes.

Tags start with "<A" (hereinafter referred to as A-tag) and tags start with ""<P" (hereinafter referred to as P-tag) accompany subsequent texts, and end with "</A>" and "/P>" respectively. Conversely, in basic language data, texts only appear right after an A-tag or a P-tag.

Fig. 6 is a diagram showing an example of the body of basic language data. In this example, a total of three (3) texts such as "select the menu" "E-mail", and "BBS" are included. On the third line, a file name of image data to be displayed is described after "SRC=". Link destinations are written after "HREF=" on the fourth and the fifth line. For example, it is apparent from the description on the fourth line that the description is linked to "MAIL.KSP" shown in Fig. 8.

(3) Style sheet

Fig. 7 is a diagram showing an example of a style sheet for HTML corresponding to the body shown in Fig. 6. The description "K1. NAME" located between "$"s on the second line shows that the description on this line indicates a display format of which one of the lines in Fig. 6. The description "K1" shows the name of the tag, so that the first line shown in Fig. 6 corresponds to the description. Interpretation of the style sheet will be described later.

3. Processing performed by the content server 30

(1) Overall processing

Fig. 9 is a flowchart of the processes performed by the content server 30. When an access from one of the terminals is detected (step S1), the content server 30 identifies the type of the accessing terminal (step S2). Another determination whether or not the accessing terminal is controllable for transition, is made depending on the type of the terminal detected (step S3).

When the terminal device is not a controllable one for transition, data written in the basic language (hereinafter referred to as basic language data) in the content requested by the terminal is converted into data in a language which can display on the terminal, and control of transition is performed based on information about session (hereinafter referred to as session information) which is generated in accordance with the content being transferred (step S4). At that time, if the size of that content exceeds the limitation of the terminal device, which is determined, based on its display restrictions, session information is generated while dividing the content into appropriate sizes. No division of the content is carried out if the size of the content not exceed the limitation of the terminal device.

When the terminal device is a controllable one for transition, basic language data in the content requested by the terminal is converted into data in the language which can display on the terminal (step S6). Further, the content is divided into appropriate sizes and the divided contents are sent to the terminal device when the size of that content exceeds the limitation of the terminal device based on its display restrictions (step S7). Neither of division of the content nor transmission of divided content is performed when the size of the content not exceed the limitation of the terminal device which is determined based on its display restrictions.

As described above, content can be provided appropriately to any

terminals regardless of its type either in controllable one or not by just preparing in the content server 30 contents written in the basic language. Furthermore, contents stored in the content server 30 can be standardized regardless of the limitation of display restrictions for each terminal.

(2) A program for identifying the type of terminals (step S2 in Fig. 9)

Fig. 10 is a flowchart of a program for identifying the type of terminals. When a terminal makes an access to the server 30, an HTTP header is sent thereto from the terminal. The server 30 identifies that the terminal uses HTML when "X_JPHONEMSNAME" is described in the HTTP header (step S11). When "X_UP_SUBNO2" is described in the header, the server judges that a language for the WAP is used for the terminal (step S12). The server 30 judges that terminal uses C-HTML (Compact HTML) when the description of "USER_AGENT" starts with "DoCoMo" (step S13). When the description of "USER_AGENT" starts with "PDXGW", the server 30 judges that the terminal uses a language for PmailDX (trademark) (step S14).

When none of above judgements are made, the server 30 judges that the terminal accessing thereto is a PC (step S15). Furthermore, the server 30 judges which one of communication methods such as personal computer communication services, e-mail, HTML, is used in accordance with the description of "USER_AGENT". The result of the judgement is stored in the memory 36 (see Fig. 2b).

As a result of the above judgements, terminals using: MML (Mobile Markup Language), language for WAP, C-HTML (compacted HyperText Markup Language), HTML, can be identified as controllable ones for transition. In the contrary, terminals using: a language for PmailDX (trademark), personal computer communications service, and an electric

mail, can be identified as uncontrollable ones for transition.

(3) Program for generating information on sessions and for controlling transition (step S4 of Fig. 9).

Subsequently, Fig. 11a shows a series of processing for generating session information and controlling transition. The processing will be performed when an access is made by the PHS terminal 8 because the accessing terminal is identified as a terminal not controllable. The details of the processing is described assuming that "MENU KSP" shown in Fig. 8 is provided by a terminal device.

Initially, a session ID is obtained at step S51 of Fig. 11a. "a session ID" is meant to refer to an identifier virtually assigned to a terminal which is under access to a content server. In this embodiment, a session ID is obtained in a way depicted in Fig. 12.

Subsequently, session information identified by the session ID is acquired from the memory 36 (step S52 of Fig. 11a). No session information corresponding to the session ID is found when the session is newly initiated. In that case, the step proceeds to step S57.

At step S57, another judgement in which designation in page is included in a request from the terminal, is made. In this example, a specific page such as "MENU.KSP" is required so that the name of the designated page is set a position of the current page (step S58 of Fig. 11a). When no page is designated, a predetermined page (such as INDEX.KSP) is set a position of the current page (step S59 of Fig. 11a).

Subsequently, the contents of the designated page are read out from the hard disk 38 and are converted into a format appropriate to the terminal (step S60 of Fig. 11a). For example, Fig. 13 shows a result of

the conversion for the inquiry shown in Fig. 6.

The data displayed in Fig. 13 is meant that the display shown therein transits respectively to a display for "e-mail" and that for "bulletin board" when respective input of numeral "1" and numeral "2" is carried out at the terminal for a requirement in input "Select the menu".

Subsequently, the content is divided into multiple portions when the size of the content exceeds display restrictions of the terminal device (step S61 of Fig. 11a). Fig. 11c shows a flowchart of a program for performing division of content.

In the process, a number BC representing the number of bytes for the converted text is counted (step S101 of Fig. 11c). Then, the number BC and limitation in size LIMIT representing display restrictions of a terminal is compared with each other (step S102 of Fig. 11c). As depicted in Fig. 11d, the limitation size LIMIT is able to distinct in accordance with the type of the terminal device identified in step S2 of Fig. 9.

The limitation size LIMIT becomes 1,000 bytes because a PHS terminal is in use. In the contrary, no division of the content is carried out and the process shown in Fig. 11c is terminated because the converted text does not exceed 1,000 bytes.

Subsequently, the content being selected is transmitted to the terminal device (step S62 of Fig. 11a). Following the transmission, session information that is necessary to perform transition control at the server is generated and stored in the memory 36 (step S63 of Fig. 11a). Fig. 14 shows session information that is generated in accordance with the content shown in Fig. 6. "session ID" in the figure is the ID

assigned according to the method of Fig. 12. "time for terminating session" is the time at which session is terminated by force, and it is set for preparing a case in which the network is disconnected without termination process and so on. The time for terminating session is set at a value, which add a predetermined period (e.g. 30 minutes) to the current time. "type of input values" shows the type of data capable of being input on pages currently transmitting (the pages displayed on the terminal device). The value of which is "3" in Fig. 14.

Characters, status and response, to each of inputs are stored in the session information. For example, "1" shows characters for input 1, and "normal" and "transit to MAIL.KSP" respectively representing status and response for that input. Characters for input 3 is "*", and that is a wild card showing "any other characters". Such wild card corresponds to characters except for both input 1 and input 2 ("1" and "2"). The status of such input is in "error" and shows "transit to MAIL.KSP" is carried out with a display of error.

As a result of carrying out such operation, a screen image depicted in Fig. 15 is displayed on the PHS terminal 8.

The process, which will be performed when "1" is input to the PHS terminal 8 under the displayed screen image, will be described hereunder. In this case, the process also proceeds to step S51 of Fig. 11a through steps S1, S2 and S3 of Fig. 9. The process further proceeds to step S54 of Fig. 11a because it is judged that session information (Fig. 14) having a session ID obtained at step S51 exists. At step S54, the character "1" input through the terminal is fetched.

The content to be displayed subsequently is determined by referring the session information of Fig.14 in accordance with the input "1" by the

user (Step S55 of Fig. 11a). Transition to data represented by "response to input 1"" MAIL.KSP" which correspond to "characters of input 1" "1" in the session information is carried out. In other words, data represented by MAIL.KSP (see Fig. 8) is read out from the hard disk 38 as the current page.

Conversion in format of the data MAIL.KSP thus read out is performed (step S60 in Fig. 11a). Fig. 16 shows the result of the conversion. No division of content is carried out and the content is transmitted to the terminal because the size of the content does not exceed the limitation in size LIMIT (steps S61 and S62 of Fig. 11a).

Following the transmission, new session information depicted in Fig. 17 is generated and stored in the memory 36 (step S63 of Fig. 11a). The previous session information is discarded. A screen image shown in Fig. 18 is displayed on the terminal device as a result of carrying out steps described above.

In the steps described above, no division of content is needed. In the contrary, the steps, which will be described hereunder, require division of content. The flowchart showing the overall process of such steps is the same as Fig. 9. Fig. 11b is a flowchart illustrating details of the steps shown in Fig. 11a.

Description of steps performed when the PHS terminal 8 requests the content (name of the file; DINING.KSP) shown in Figs. 19a, 19b will be made. In this case, the PHS terminal 8 sends to the content server 30 a request such as @KCCS/DINING.KSP. In other words, the request standing for KCSS shows the address of the server, and DINING.KSP representing name of the file.

Steps S1 through S3 of Fig. 9 to be performed herein are exactly the same ones in the previous description. The step proceeds to step S4 because the PHS terminal 8 is judged as a terminal uncontrollable one for transition. Fig. 11b illustrates details of step S4. The process proceeds to step S83 through steps 81 and S82.

At step S83, the session currently carrying out is judged as new one, and then the process proceeds to steps S92 through steps S89 and S90. Conversion in format for the PHS terminal 8 is carried out at step S92. The resultant content is shown in Figs. 20a, 20b.

Subsequently, division of content is performed at step S93. Fig. 11c shows details of the process performed therein. At step S101, a number BC representing the number of bytes for the converted text is counted. The number BC in an example shown in Figs. 20a, 20b is 1,875 bytes.

According to Fig. 11d, limitation in size LIMIT for the terminal is 1,000 bytes. At step S102, the number BC and limitation in size LIMIT is compared with each other (step S102 of Fig. 11c). In this case, the process proceeds to step S103 for preparing division of content because the number BC exceeds the limitation LIMIT.

Preferably, text consist of paragraphs is divided on a paragraph basis in view of its readability. To achieve that, a maxim paragraph position PDIV until which paragraphs not exceeding the LIMIT being included is detected while searching segments of the paragraphs, that is codes for line feed (step S103 of Fig. 11c). The number of bytes till the seventh paragraph (start with "The whole concept of the program...so on " and ends "both hands grab the forks") is 968 bytes, and that by the eighth paragraph (start with "This algorithm easily so on" and ends with

"this is a typical example so called "dead lock"") is 1,315 bytes. This result in PDIV equal to 986 bytes.

In an exceptional case, PDIV could be equal to zero when a long paragraph(s) is included in the resultant text thus converted. In that case, segments of the paragraphs by which the value of PDIV being in one or more are determined by narrowing unit for searching from a paragraph basis to sentence basis, even to a character basis.

Subsequently, data of the converted text until PDIV-th byte is stored in the buffer in order to transmit it to the terminal device (step S104 of Fig. 11c). A prompt "(1) next" by which divided pages following thereto being selected are recorded in the buffer (step S105 of Fig. 11c). As a result of this, content shown in Fig. 24 is stored in the buffer.

Session information is updated along with division of the text. Depicted in Fig. 21 is session information generated based on the content illustrated in Figs. 19a, 19b. Items following an item "type of input value" within the session information are backed-up (step S106 of Fig. 11c). Then, each of the columns corresponding to items such as type of input value, input value 1, and input value 2 is edited as shown in Fig. 22 (steps S107, S108, and S109 of Fig. 11c). A part of the text following the PDIV is stored by inserting a column "subsequent text" into session information (step S110 of Fig. 11c). This enables the server to prepare a request for displaying a subsequent page(s).

In this embodiment, session information depicted in Figs. 23a, 23b is generated as a result of performing the above process.

Subsequently, the process returns to step S94 of Fig. 11b in which the content stored in the buffer (Fig. 24) is transmitted to the terminal.

In this way, a screen image identical with the display depicted in Fig. 24 is displayed on the PHS terminal 8.

Description of steps performed when "1" is input through the PHS terminal (in other words, a request for displaying a divided subsequent page(s)), will be made. On inputting such an input, the process proceeds to step S4 steps through S1, S2 and S3. The process proceeds to step S83 through steps S81 and S82 in Fig. 11b illustrating the details of step S4.

At step S83, the session currently carrying out is not judged as new one, so that the process proceeds to steps S84. At step S84, the input performed by the user is fetched. In this case, the character "1" input through the terminal is fetched. Subsequently, a page being displayed is determined in accordance with session information and the inputted character "1" (step S85 of Fig. 11b). Content of the subsequent text depicted in Figs. 23a, 23b is the one being displayed.

A judgement whether or not the one being displayed is the display of a divided subsequent page(s) (step S86 of Fig. 11b). The process proceeds to step S87 because the displayed one is a divided subsequent page in this case. At step S87, the session information backed-up previously at step S106 of Fig. 11c is restored (step S87 of Fig. 11b). By performing these processes, the session information is restored as depicted in Fig. 21.

Subsequently, another processing for dividing content shown in Fig. 11c is performed using the content of the subsequent text as the one being displayed (step S88 of Fig. 11b). No division of content is carried out because the size of the content does not exceed the limitation in size LIMIT. Thereafter, the text to be displayed is determined (step S89 of

Fig. 11b). In this case, exactly the same one to the content of the subsequent depicted in Figs. 23a, 23b is displayed on the PHS terminal 8.

As described above, the content server 30 in accordance with the present invention can respond on a real time basis to the display restrictions of the terminal devices so as to perform proper display thereon. In this way, the server 30 just stores common contents for proper display regardless of the display restrictions of the terminal devices.

(4) A program for converting a language into another language (step S5 of Fig. 9)

Subsequently, processing, performed when a terminal accessing is controllable for transition, will be described hereunder. In such a case, the process proceeds to step S5 from step S3. A program shown in Fig. 25 for converting a language into another language is performed at step S5. At the step, processing for converting content written in the basic language into data written in the display language for the terminal is carried out.

At step S41 of Fig. 25, data of the content written in the basic language is converted into data having a tree structure. Subsequently, the content is converted into data written in a desired display language in accordance with the tree structure data (step S42 of Fig. 25).

The conversion processing, in which the PC 2 using a browser software makes a request for viewing content written in the basic language, which includes both the style sheet shown in Fig. 7 and the body depicted in Fig. 6, will be described. In this case, the display language used for the terminal can be judged as HTML (see Fig. 10).

Hence, the language used for that content need to be converted into HTML eventually.

The processing for converting the basic language data into data having a tree structure (step S41 of Fig. 25) will be described. In order to perform such processing, tags on each line of the body depicted in Fig. 6 are analyzed into the data structure shown in Fig. 26. With regard to subsequent tag number and tag number for child-tags will be described below.

The body depicted in Fig. 6 may express in a tree structure as illustrated in Fig. 27. In the drawing, the numerals within in parentheses indicate tag number. The drawing shows that the entire data of KSP consist of P-tag, IMG-tag, A-tag, and another A-tag.

The relationship shown in Fig. 27 may also be expressed as Fig. 28. In other words, the relationship can be expressed using these links, a parent-to-children link indicated in view of parent tag and child-to-child links (brother/sister links) indicated in view of a highly classified child to a lower classified child having the same parent. In this embodiment, the relationship is expressed in a method shown in Fig. 28. It is, therefore, the body depicted in Fig. 6 is converted into data shown in Figs. 29 and 30 having a tree structure.

Upon creation of the tree structure data as described above, the original data is converted into data written in the display language using this tree structure data (step S42 of Fig. 25). Flowcharts of such conversion processing are shown in Figs. 31, 34a and 34b. Fig. 31 shows converting process in which the display language after the conversion accompanies a style sheet and Figs. 34a, 34b illustrate another conversion process in which the display language after the conversion

does not accompany a style sheet.

This description assumes the conversion into HTML data so that the conversion is performed according to a display format described in a style sheet because HTML data always accompanies a style sheet. Therefore, a program for converting basic language data into HTML data shown in Fig. 31 is performed as described below.

Characters in the style sheet depicted in Fig. 7 are read out sequentially and the characters thus read out until the character "$" are copied on an output file (written in HTML) (step S21 of Fig. 31). In the case of using the style sheet depicted in Fig. 7, all the characters "<HTML> line feed <HEAD><TITLE> " are copied on the output file. Subsequently, a judgement whether it is the end of the file (end of the style sheet) (step S22 of Fig. 31). When it reaches to the end of the file, the process ends thereat.

Else, the identifiers (a series of alphanumeric) come subsequent to "$" in the style sheet are read out therefrom. The alphanumeric is stored in the memory 36 as a parameter KEY because the alphanumeric represents a key name assigned to a tag of the basic language data (step S24 of Fig. 31).

Subsequently, the tag having a key indicated by the parameter KEY is searched from the tree structure data shown in Figs. 29 and 30 and store the tag number thereof into the parameter TAG (step S24 of Fig. 31). Here, the value of the parameter TAG is in 1, because tag number 1 has the key name K1.

Next, another judgement, whether or not the subsequent character on the style sheet is a period, is carried out (step S25 of Fig. 31), if the

upcoming character is judged as a period, the content server skip the period from reading, and then identifiers (a series of alphanumeric) subsequent to the period are read out. The alphanumeric thus read out substitutes for a parameter ATTR (step S26 of Fig. 31). Thereafter, the value of an attribute having attribute name ATTR is copied on the output file from a tag indicated by tag number TAG (step S27 of Fig. 31). Here, the attribute name ATTR is equivalent to an identifier NAME because the attribute come after the period is the identifier NAME. Thus, the value of the attribute NAME in the tag number 1 such as "KSP sample " is output on the file.

When a judgement in which the subsequent character is not a period, is made at step S25 of Fig. 31, the text column out of a tag indicated by TAG is copied on the output file (step S30 of Fig. 31).

Subsequently, another judgement, whether or not the subsequent character is "$", is carried out (step S28 of Fig. 31), the content server skip reading it when the subsequent character is "$"(step S29 of Fig. 31), and then the process goes back to step S21. Else, the server judges that an error occurred and ends the process.

Fig. 32 shows the HTML data thus converted according to the above description. Fig. 33 shows a screen image displayed on the screen of the PC 2 with the HTML data shown in Fig. 32. Wallpaper 50 is displayed with the tag located on the third line. The menu is displayed thereon with the tags located on the fourth, sixth, and seventh lines. It is apparent from the displayed image that images "E-mail", and "BBS" being underlined are linked in response to tags on the sixth and seventh line. Further, another image 52 is displayed with the tag located on the fifth line.

As described above, the use of a style sheet let the server obtains pure HTML data. In addition, the style sheet allows these in display: decorative work such as creating tables, centering, right-justify, blinking, as well as banner advertising, icons.

Although, data conversion in language from the basic language to HTML has been described in the above description, programs for converting basic language data written into data in other display languages are stored in the server so that a conversion program appropriate to a terminal will automatically selected and used depending on the type of the terminals.

Subsequently, processing performed by the server when a request for viewing a content stored therein is output from the WAP terminal 6 (a mobile phone) will be described. In this case, the algorithm shown in Fig. 10 is also used for identifying the type of a terminal. With this algorithm, it is known that the basic language data need to be converted into HDML data.

Exactly the same processing to the processing described in the above is performed until converting the basic language data into data in a tree structure as shown in Fig.27. However, the program shown in Figs.34a, 34b is executed because no style sheet for HDML data exist in the basic language data illustrated in Figs. 6 and 7.

In the program, tag number of the tag located at the beginning of the tree structured data shown in Figs. 29 and 30 is acquired and is substituted for a parameter TOP (step E1 of Fig. 34a). Here, the value of the parameter TOP is equal to 1 because an element located at the beginning thereof is tag number 1. Then, the type of tag indicated by the parameter TOP is fetched and is substituted for a parameter KIND

(step E2 of Fig. 34a). Here, it is, therefore, the parameter KIND is equivalent to "KSP".

Thereafter, the process is branched depending upon contents of the parameter KIND (step E3 of Fig. 34a). Here, the process proceeds to step E4 because the parameter is equal to "KSP".

At step E4, characters "<HDML VERSION=3.0><line feed> <DISPLAY>"are output as header information of HDML on the file. Then, the tag number of a child tag for the TAG number 1 is set as the beginning of the tree structured data, and the processing shown in Figs. 34a 34b is recalled recursively (step E5 of Fig. 34a). In this case, the parameter TOP comes to 2 at step E1 and the parameter KIND is in "P" at step E2.

Thus, the process proceeds at steps E24 and E25 from step E3 through steps E7, E15. At step E25, contents of the text column of a tag, its tag number being TOP, is output as it is. Here, a text column "select the menu" is output to the file. Subsequently, a tag "<BR>" is output for line feed. (step E26 of Fig. 34b).

Next, the process proceeds to step E27, and the value of a column "subsequent tag number" of the parameter TOP is substituted for the parameter TOP. Here, the value of the parameter TOP is equal to 3. Subsequently, a judgement whether or not the parameter TOP is equal to 0 is carried out (step E28 of Fig. 34b). In this case, the process returns to step E2 because the parameter TOP is not 0.

Thereafter, the type of a tag indicated by the parameter TOP is identified at step E3. In this case, the process proceeds to step E8 from step E7 because the parameter KIND is equivalent to "IMG". At step

E8, an HDML tag indicating images <IMG SRC> is output (step E8 of Fig. 34b). Subsequently, the name of an image file is obtained by using the SRC attribute of a tag indicated by the parameter TOP and the SRC attribute is substituted for a parameter FILE (step E9 of Fig. 34b). In this case, the parameter FILE is equal to "ICON. JPG". Then, steps following step E11 are executed when the parameter FILE is not "BMP" as a result of carrying out a search for a file extension of a file indicated by the parameter FILE (step E10 of Fig. 34b). On the contrary, the process proceeds to step E13, when the file extension is in "BMP". In this case, the steps following with step E11 of Fig. 34b will be executed because the extension is "JPG" indicating JPEG images, not "BMP".

Images in "ICON. JPG" are converted into images in a monochrome BMP format at step E11 of Fig. 34b. Next, the extension of the parameter FILE is converted into "BMP" from "JPG"(step E12 of Fig. 34b). It is, therefore, the parameter FILE becomes "ICON. BMP".

Subsequently, characters indicated by the parameter FILE are output (step E13 of Fig. 34b). Further, an ending tag of the images and a tag for line feed "></BR>" are output (step E14 of Fig. 34b).

Thereafter, the value of the parameter TOP is updated (step E27 of Fig. 34b). In this case, the parameter TOP becomes 4 so that the process proceeds to step E2 through step E28, and the parameter KIND is equal to A thereat. Thus, the process proceeds to step E16 through steps E3, E7 and E15.

At step E16, an HDML tag "A TASK=GO DEST=" representing hyper-link is output to the file (step E16 of Fig. 34b). Subsequently, an HREF attribute of a tag indicated by the parameter TOP is looked over and the value of the attribute is output (step E17 of Fig. 34b). In this

case, the value thus output is "MAIL.KSP".

Next, after a tag "ACCESSKEY=" is output (step E18 of Fig. 34b), the value of a parameter AK is incremented by 1 (step E20 of Fig. 34b). Here, the tag ACCESSKEY is a tag for accelerating operation performed with numeric keypads on the mobile phones, generally a series of numerals used more than one.

Thereafter, the A-tag is finalized by outputting a character ">" (step E21 of Fig. 34b), and the text column of a tag indicated by the parameter TOP is output (step E22 of Fig. 34b). In this case, "mail 1" is written in the text column. Then, the A-tag is closed with characters "</A>
" (step E23 of Fig. 34b).

As described above, HDML data, the resultant data of conversion, is obtained as an output file. Thus, conversion of display language into the standard display format is performed when no style sheet for its display language exists in the basic language data. HDML data, the resultant data, is illustrated in Fig. 35. Fig. 36 shows an example of an image of the HDML data displayed on the screen of the WAP terminal 6.

Although, the conversion into HDML data from the basic language data have been described in the above, other programs for language conversion into other display languages are stored in the server so that an appropriate program for conversion is automatically selected and used depending on the type of the terminal accessing the server.

The basic method of converting a language used by an accessing terminal into anther display language employed in the programs is similar to that described in the above. But no function of requiring a link destination is equipped with the terminal when the terminal use a

protocol for E-mail and the terminal is used for personal computer communications service. Thus, the current status of the terminal and the link destination after the selection (the transition destination) need to be managed by the server 30. In order to fulfill this requirement, it is necessary for the server not only creating display language data but also managing the status of the terminal, and to create data that can be used for determining the link destination when the link destination is described in the basic language data.

In this embodiment, the conversion is performed on a real time basis whenever the terminal requires it. In this way, update of the content can immediately reflect of the display thereof on the terminal whenever the content is updated

In this embodiment, content data written in the basic language is stored in the server, and the data is converted into display language data written in a language appropriate to the type of a terminal accessing to the server. In this way, content can be viewed on every terminal by just preparing contents written in the basic language.

In above the embodiments, descriptions, regarding the cases in which no division of the content is carried out, have been made. Description, regarding the case in which division of the content is carried out because of the restrictions of the terminal device, will be made.

Assuming that a terminal device requests a content depicted in Figs. 19a, 19b. The details of steps S1, S2, S3 and S6 of Fig. 9 are the same as that of described in the above description. Description will be made on the assumption that the terminal device uses HTML and the limitation size LIMIT thereof is 1,000 bytes. The limitation size LIMIT is determined depending on the type of terminal devices.

At step S5 of Fig. 9, the content is converted into HTML data depicted in Figs. 38a, 38b. Fig. 37 shows the details of processing for carrying out division.

In the process, a number BC representing the number of bytes for the converted text is counted (step S201 of Fig. 37). Then, the number BC and limitation in size LIMIT representing display restrictions of a terminal is compared with each other (step S202 of Fig. 37). In this case, the process proceeds to step S203 of Fig. 37 because the number BC is larger than the limitation LIMIT.

At step S203, the content is divided into a group of paragraphs, each does not exceeds the limitation LIMIT. Further division on a sentence basis or even on a character basis will be carried out when the number of bytes for one paragraph exceeds the limitation LIMIT.

Subsequently, temporary based Uniform Resource Locators (URLs) are assigned to each of the divided paragraphs (step S204 of Fig. 37). Figs. 39 and 40 shows the content thus divided in this way.

A temporary URL (DINING_2.HTML) for the subsequent content is described as the last link information. This forms a link between the divided content.

After dividing the content in such a way, the content depicted in Fig. 39 is transmitted to the terminal device at step S205 of Fig. 37.

As described above, contents to be prepared in the content server 30 can be used commonly to all the terminal devices regardless of the display restrictions of the terminal devices.

Link information (<A HREF="DINING_2.HTML">) forms a link between the divided content depicted in Figs. 39 and 40 as a result of carrying out division of content. However, transition between the divided content may be controlled at the server 39 with session information generated shown in Figs. 23a, 23b without using link information created by the division.

4. Other embodiments

(1) The processing for converting content is performed within the content server 30 in the embodiments described above. Alternatively, both a content server 120 storing the content written in the basic language and a converting server 110 storing a program for the conversion and performing conversion processing may be provided independence of each other as shown in Fig. 41.

In order to distribute the load (or providing a backup server) as a result of independently providing the servers, the administrator of the system just provides two of the converting servers 110 so that the administrator is free from preparing two identical contents. In this way, maintenance of the contents can be simplified.

(2) Although, the conversion is performed on a real time basis in the descriptions stated above, content which have been converted with the program for conversion may be provided. Alternatively, the conversion may be performed on a real time basis in accordance with the data converted into the tree structure shown in Figs. 14 and 15, which have been stored in the server.

(3) The type of the terminal (both in hardware and software) accessing is identified by HTTP header in the embodiments described

above. Such identification may also be done with its IP address.

(4) In the embodiments described above, content written in the basic language is stored, and such content is converted into content written in a display language for the terminal. However, the content written in a desired display language can be obtained by performing the processing with a system shown in Fig. 42 in which all the display languages for the terminals are converted once into the basic language using a first converting device 140 storing a first conversion program and then the converted content in the basic language is converted into the content written in the desired language using a second conversion device 130 storing a second conversion program. In this way, contents written in any display language can be viewed on all the terminals.

In addition, both the converting devices 130 and 140 may perform the conversion on a real time basis, or they may store the resultant data after performing the conversion thereby.

(5) Displays on the screen have been used as an output format of the content in the embodiment described above, the present invention can also be applied to other output formats such as an audio based output. For example, content can be divided into multiple content similar to the embodiment described above when the content has a limitation in size because of the restrictions on audio outputs at terminal devices.

(6) Division of content is carried out after storing the content and converting language thereof, which is written in the basic language, into another language. Such division of content may be carried out by storing content written in languages for each of the terminal devices and then dividing the content into multiple content each having a certain size so as not to exceed the limitation in size for each terminal device.

While the embodiments of the present invention, as disclosed herein, constitute preferred forms, it is to be understood that each term was used as illustrative and not restrictive, and can be changed within the scope of the claims without departing from the scope and spirit of the invention.

What is claimed is:

1. A content server receiving an access by different types of terminal devices comprising:

content storage means, storing content;

terminal identifying means, identifying a display restriction of the terminal device accessing to the server; and

data format adjustment and transmission means, adjusting a data format of the content read out from the content storage means so as to match a display restriction of the terminal device depending on the display restriction of the terminal device and transmitting the adjusted content to the terminal device.

2. A program for realizing a content server receiving an access by different types of terminal devices with a computer, the content server comprising:

content storage means, storing content;

terminal identifying means, identifying a display restriction of the terminal device accessing to the server; and

data format adjustment and transmission means, adjusting a data format of the content read out from the content storage means so as to match a display restriction of the terminal device depending on the display restriction of the terminal device and transmitting the adjusted content to the terminal device.

3. A storage medium to store the program defined in claim 2.

4. The content server of claim 1 wherein the display restriction is a size of content which can be stored temporarily at the terminal device for its display,

and wherein the data format adjustment and transmission means divides the content in accordance with the size of the content.

5. The content server of claim 4 wherein the data format adjustment and transmission means generates relevant information between the divided content when the content is divided, and wherein the relevant information is sent to the terminal device.

6. The content server of claim 1 wherein the terminal identifying means detects the display restriction of the terminal by identifying a type of the terminal device.

7. The content server of claim 6 wherein the terminal identifying means identifies the type of the terminal device in accordance with a header of Hyper Text Transfer Protocol (HTTP) sent from the terminal.

8. The content server of claim 6 wherein the terminal identifying means identifies the type of the terminal device in accordance with an IP address sent from the terminal device.

9. A content server receiving an access by different types of terminal devices comprising:

content storage means, storing content; and

control means, controlling read out the content and transmission thereof on receipt of access from the terminal device;

wherein the control means identifies a display restriction of the terminal device accessing to the server and adjusts a data format of the content read out from the content storage means so as to match a display restriction of the terminal device depending on the type of the terminal device and transmits the adjusted content to the terminal device.

10. A program for realizing a content server receiving an access by different types of terminal devices with a computer, the content server

comprising:

content storage means, storing content; and

control means, controlling read out the content and transmission thereof on receipt of access from the terminal device;

wherein the control means identifies a display restriction of the terminal device accessing to the server and adjusts a data format of the content read out from the content storage means so as to match a display restriction of the terminal device depending on the type of the terminal device and transmits the adjusted content to the terminal device.

11. A storage medium to store the program defined in claim 10.

12. A data format adjusting device for adjusting a data format of content, wherein the device is designed to receive a display restriction of a terminal to which the content is transmitted and to adjust the data format of the content read out from content storage means so as to match the display restriction of the terminal device depending on the display restriction thereof.

13. A program for realizing a data format adjusting device for adjusting a data format of content with a computer, wherein the device is designed to receive a display restriction of a terminal to which the content is transmitted and to adjust the data format of the content read out from content storage means so as to match the display restriction of the terminal device depending on the display restriction thereof.

14. A storage medium to store the program defined in claim 13.

15. The data format adjusting device of claim 12 wherein the display restriction is a size of content which can be stored temporarily at the terminal device for its display, and wherein the adjustment is

performed by dividing the content in accordance with its size.

16. The data format adjusting device of claim 15 wherein the device generates relevant information between the divided content when the content is divided.

17. A method of distributing content to different types of terminal devices, the method comprising the steps of:

storing the content;

detecting an output restriction of the terminal device accessing thereto;

adjusting a data format of the content so as to match the output restriction of the terminal device depending on the output restriction of the terminal, and

transmitting the adjusted content to the terminal device.