Web page Information Architecture Formalization Method and It's an Example

Yorinori KISHIMOTO

***Abstract*—This paper proposed a formalization method of Web page information architecture by regular expression for checking its structure. This method classifies structure elements of a Web page in attributes on the basis of an idea of the Web information architecture, and it's expressed by two types of equations on the basis of the F-Shaped reading pattern. This method can verify structure of the information architecture of a Web page. As a result, this method was able to analyze about structure elements of lack point and a redundant point in Web page information architecture.**

***Index Terms*—Web information architecture, Formalization, Regular Expression.**

I. Introduction

THE user interface quality is important things of Web site. A common method of checking interface quality is a questionnaire result of completely Web site. However, this method is sensory evaluation. So, this method is difficult to show clearly the quality of Web site as a numerical value. In addition, this method can apply to only after a Web site development work completion. So, this method cannot apply to the phase of designing a Web site.

For analyzing performance of Web site, a point of view of the information architecture has been proposed[1]. It considers from the structural side in information elements composing Web site, and this idea derives information structure of a high usefulness for a user of Web site. However, it is required that experience of many Web designs and interface designs because this idea includes many different concepts to use it by a Web design work. In addition, it is difficult to use this idea in a practical field, because the method to express clearly the Web information architecture is few. Thus, we proposed a formalizing method for express information architecture of a Web page by regular expression.

According to the idea of the information architecture, structure elements of a Web page can be classified roughly into the browsing support, the search system, the contents task and other invisible elements. In addition, it had been proposed that eyes transition of Web site reader becomes an F-shaped reading pattern generically. Then we focused on information architecture elements and eyes transition of Web site users. It is required that the information architecture of a Web page can browse a user without of the efficiency and inconsistency. In other words information structure quality of a Web page can be argued about if based on this structure.

Tokyo University of Information Sciences Department of Information Systems, 4-1 Onaridai, Wakaba-ku, Chiba, 265-8501 Japan e-mail: kisimoto@rsch.tuis.ac.jp

In this method, Web information architecture is expressed by regular expression. At first, information architecture construction elements are abstracted as a variable based on an attribute of the Web information architecture. In this abstraction, a meaning of sentences and/or design elements these are included in a construction element of a Web page can be ignored, and only a meaning as a structure element can be extracted. It can be expressed as a deterministic finite-state automaton by these abstracted elements concatenated along an F-shaped reading pattern of Web site user.

This paper proposed that a method to formalize the information architecture in a Web page by regular expression. In addition, it was confirmed that this example could apply this method. As a result, total information architecture of Web page could be expressed by regular expression, and it has been detected to a lack point and redundant point in this example's information architecture.

II. Ideas

A. Program Structure Formalizing method

Web page information architecture formalizing method based on an idea of the program structure formalizing method. So, here shows an idea of the program structure formalizing method.

A program structure is derived from a program by neglecting any kind of proposition in it.

If a program is well-structured, regular expressions can be applied to represent the program construct sequence with three kinds of constructs, in which the concatenations are represented by ·(AND), the selections by +(OR), and the iterations by Kleene closures $(^*, ^+)$. Some program constructs could have relations to others in a program context. In this case, a program does not always keep the context in it in handling equations regularly. To resolve the problem of context linkages, the concept of ' the **program structure** and the **program mechanism**' have been introduced[2]. The structure is non-deterministic in expressing the frame of a program, while the mechanism puts a constraint on the production of a specific program from a structure. In other words, a program consists of two parts; the structure and the mechanism.

Formalization of the regular expressions can be applied to the program structure alone, due to its being non-deterministic. A program structure can be identified from a program by removing the mechanisms and constructing program constructs. The program structure means the maximum framework of the program which is specified by the mechanisms. Different programs could be derived by applying different mechanisms to one program structure[3]. Since the description level is higher in the abstract form than in code, it becomes close to the specifications, and is effective for analyzing the traceability of programs to their specifications and checking logics for them[4][5].

It was proposed that the program structure formalization method can analyze good and/or poor point of a program[6][7]. In this proposal, it shows this method can discuss the lack point, the redundant point and the same program structure[8](see Appendix).

B. Basic Idea

It is difficult to shown clearly a quality of Web usability. A common method of checking interface quality is a questionnaire result of completely Web site. This method can show a quality of Web usability by sensory evaluation. However, this method requires many respondents to a questionnaire. And it requires the completed Web site development. So, this method cannot apply to the development phase. And, this method is difficult to discuss about a possibility of Web page layout design.

Web information architecture had been proposed for an idea of Web usability. This idea focused on the classifications of information elements in Web pages. And this idea proposed that these elements classifications and its layout designs are very important things about Web usability. On the basis of this idea, it can discuss about Web usability because it focused on Web page elements classifications and its layout design. So, concatenation of construction elements of Web page is defined as the information architecture of a Web page.

If information architecture's structure of Web site is poor quality, a user of this Web site feels difficulty. It is necessary to considering to naming rules of links and/or classification system of a category by an idea of the Web information architecture. Especially, in a Web page designed based on a definite rule, disarray of context becomes it with a factor harming usefulness.

In addition, similar Web page elements are set to be adjacent. For example, a category list of similar items is adjacent in a design of shopping site. It derives from the classifications of Web page elements. Naturally usefulness becomes bad when these elements lie scattered. In other words quality of a usefulness of a Web page can be discussed on by focusing its attention on the concatenate of a structure element of a Web page.

In idea of the program structure formalization method can analyze on a regular expression of a program structure. It had been proposed that analysis about lack point and redundant point of the program structure that represented by regular expression. So, I focused on the stream of the Web information architecture as regular expression. Web information architecture's structure represented by regular expression means framework of Web information architecture. Because, when focus on this framework, it can be discuss about good and/or poor point of Web information architecture.

Thus, we focus on a concatenation of information architecture elements as a purpose to usefulness measurement of a Web page.

TABLE I
ELEMENTS OF WEB INFORMATION ARCHITECTURE

Category	Attribute
Browsing support	Global Navigation
	Local navigation
	Context navigation
	Site map
	Site index
	Site guide
	Other navigation
Support of search	Search interface
	Search zone
Contents task	Headline
	Embedded link
	Embedded meta data
	Chunk
	List
	Continuous step
	Identifier

C. A classification of structure elements

In the Web information architecture, a classification of Web site structure element is proposed. On the basis of this idea, information architecture elements were classified. A structure element of a Web page can disregard an invisible element such as thesaurus. In addition, this can be disregarded because a query language in a search support and search algorithm is not structure elements of a Web page element directly.

Information architecture elements in a Web page on the basis of this idea is shown in TABLE I.

D. Stream of Web information architecture

When users read Web page content, the reading pattern of them became F-shape was proposed by Jakob Nielsen [9]. Then, we focus on this reading pattern. In this case, users of Web site reads Web page elements sequentially. When this stream is not methodologically, users cannot understand how to use this Web site. Thus, the input stream of user's information is Web information architecture elements sequence.

So, Web information architecture's structure can be considered as this reading sequence of Web page elements. Fig. 1 shows image of this idea[10].

Some Web page's reading pattern is not become F-shaped. For example, a result of Web search reading pattern becomes E-shaped shown in Fig. 2[11].

In this case, a stream of Web information architecture is becomes more natural as envisioned users reading pattern.

E. Formalization of the information architecture of Web page

An information architecture structure element of a Web page was classified in TABLE I. A structure element of a Web page has a design, a color, a sentence, shape, and an attribute as a structure element of the information architecture.

However, these mixtures make analysis of Web information architecture difficult. Thus only a structure element attribute of the information architecture is extracted in a structure element and an extracted element is abstracted as a variable of regular expression.

TABLE II shows correspondence between these elements and its abstracted variables.

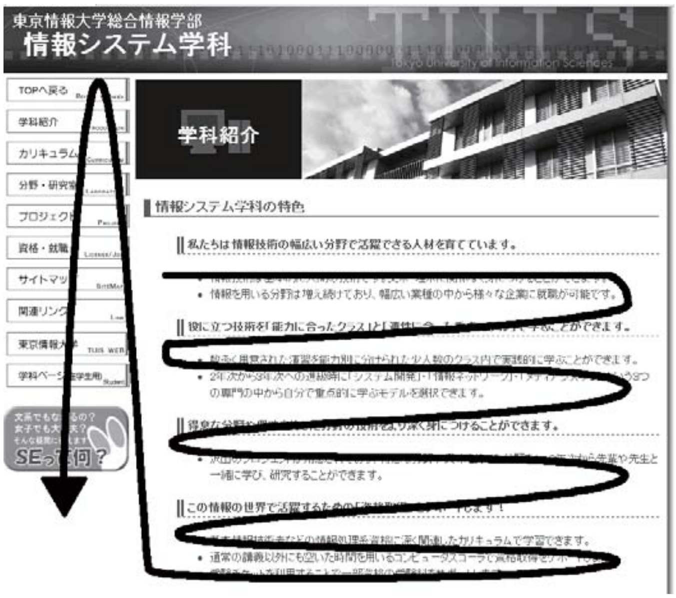

Fig. 1. Image of F-Shaped reading pattern

Fig. 2. Image of E-Shaped reading pattern

Furthermore, Web information architecture's structure become a stream of Web page elements derived from an idea of F-shaped reading pattern. This stream is concatenations of Web page elements these abstract by regular expression variables. So, Web information architecture can be shown by a regular expression.

For example, here shows simple Web page example. Elements of Web page in Fig. 3 can abstract by regular expression variables.

When this page reading pattern is F-shaped, the stream of reading pattern is shown in Fig. 4.

A regular expression is derived from concatenations of Web page elements along the F-shaped reading pattern. In this case, a regular expression derived from this example is shown in equation (1).

$$X = Cc_1 Nl_1 Nl_2 Nl_3 C_1 Ng_3 Ng_2 Ng_3 \quad (1)$$

This method can be shown clearly to the information architecture in a Web page.

TABLE II
ELEMENTS OF WEB INFORMATION ARCHITECTURE

Category	Attribute	Variable
Browsing support	Global Navigation	Ng
	Local navigation	Nl
	Context navigation	Nc
	Site map	Nm
	Site index	Ni
	Site guide	Nu
	Other navigation	N
Support of search	Search interface	Si
	Search zone	Sz
Contents task	Headline	Cc
	Embedded link	Cl
	Embedded meta data	Cm
	Chunk	C
	List	Ct
	Continuous step	Cs
	Identifier	Ci

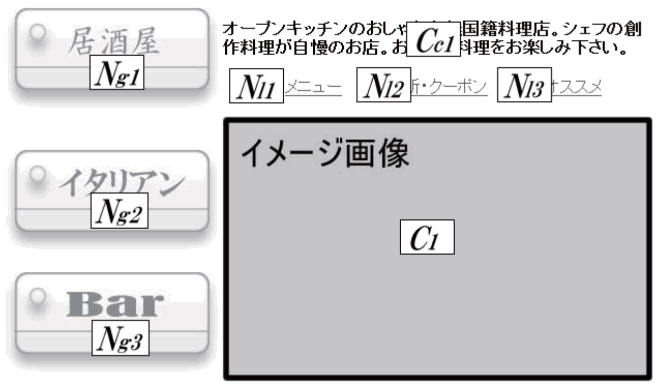

Fig. 3. Simple Web page example

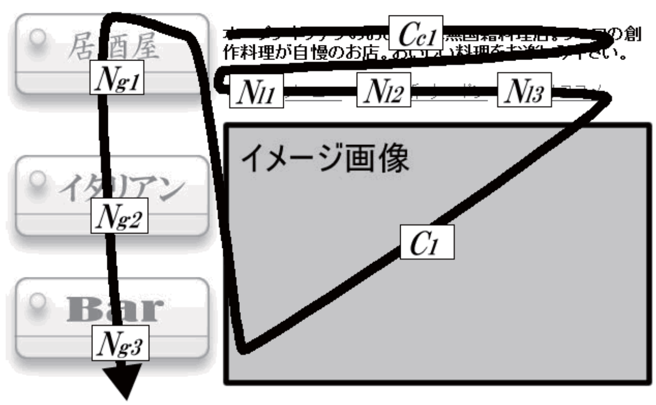

Fig. 4. Reading pattern of simple Web page example

F. An idea of analysis

In information architecture analysis of a Web page, a regular expression derived by a method of before is handled as two different expressions. One expression treats each structure element as an individual variable. In this case navigation to have a common link and/or common information is handled as the same variable.

In addition, variables have suffix for keep uniqueness.For example, Ng_1, Ng_2 and so on. By this, a structure of the whole Web page can be expressed.

More one expression is a regular expression that was provided by abstracting a variable of a structure element only by an attribute of own. In this case, suffix of variables are neglected. In this expression, concatenate of structure elements of same attribute element is expressed by Kleene enclosure iteration (* symbol). This analysis method uses both equations.

$$\begin{aligned} A &= Cc_1C_1N_1Cc_2C_2N_1Cc_3C_3 \\ &\quad N_1Cc_4C_4N_1Cc_5C_5Nl_1 \\ A' &= (CcCN)^*CcCNl \end{aligned}$$

Equation A and A' are the results that formalized the same Web page, and a meaning of a used variable name is based on TABLE II. Equation A abstracted structure elements of a Web page to an attribute and an identifier. Equation A' abstracted equation A only in attribute and expressed an iteration by Kleene enclosure. Structure of equation A' consists of two parts of $(CcCN)^*$ and $CcCNl$.

However, $(CcCN)^*Nl$ can be in this structure naturally because it can be confirmed that is one group in equation A. In other words equation $(CcCN)^*CcCNl$ is naturally the structure that should become $(CcCN)^*Nl$. And equation $(CcCN)^*CcCNl$ understands that it is the poor information structure become it lacked N.

Correspondence elements of N in equation A' is N_1 which appeared in equation A. When discuss about this problem, you focus on correspondence elements between equation A and original Web page. In this way it can be analyzed about the quality of information architecture of a Web page by analyzing two regular expression equations.

III. An analysis method

An analysis procedure by this method is shown next.

A. Step 1 : Structure elements separation

In this step, you focus on a display image of Web page that has information architecture elements such as buttons, link lists, texts, images and so on. Web page information elements classification appeared in TABLE I. So, you should be dividing Web page into information architecture elements.

More easy method of this step is printed out a Web page, and marking information architecture elements on this. For example, Fig. 5 is shown an example of this step.

Fig. 5 is Web site of Tokyo University of Information Sciences Department of Information Systems[10]. In this case, you may write squares on this for checking information architecture elements as Fig. 5.

B. Step 2 : Setting of variables

Variable names appropriate to each attribute in TABLE II and subscripts for its identification are added to structure elements divided in Step 1. An element having the same function and the same link defines it as the same variable name and subscript. For example, Fig. 6 shows an example of this step.

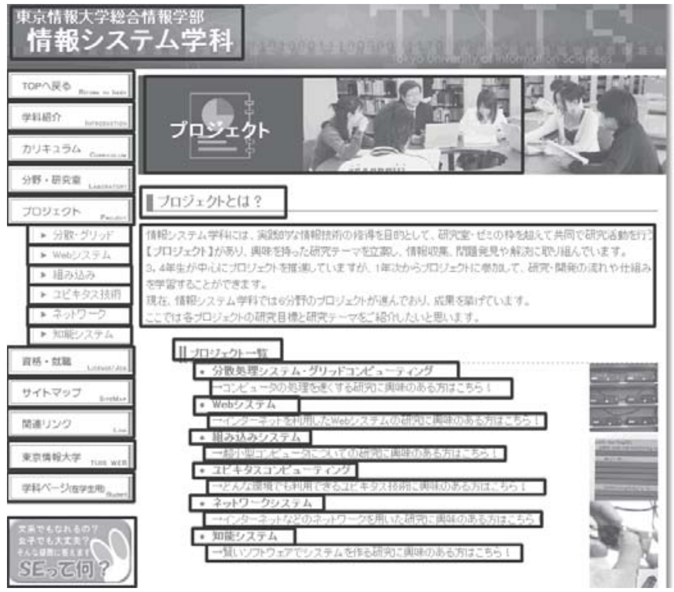

Fig. 5. Step 1 of this method

Fig. 6. Step 2 of this method

Information architecture elements are separated by squares in Fig. 6. In this case, buttons are set to same variables, because, these buttons has same category of this Web page. So, you may write the variables of information architecture elements like Fig. 6.

Variable names Nl_1-Nl_6 are two parts in this example. In this example, means of same information architecture elements described same variables.

C. Step3 : Formalization

A variable provided in Step 2 is arranged along F-shaped reading pattern. For example, Fig. 7 shows an example of this phase.

Reading pattern of this example is F-shaped. In this case, an equation of this Web page information architecture is derived from this stream of Web information architecture elements.

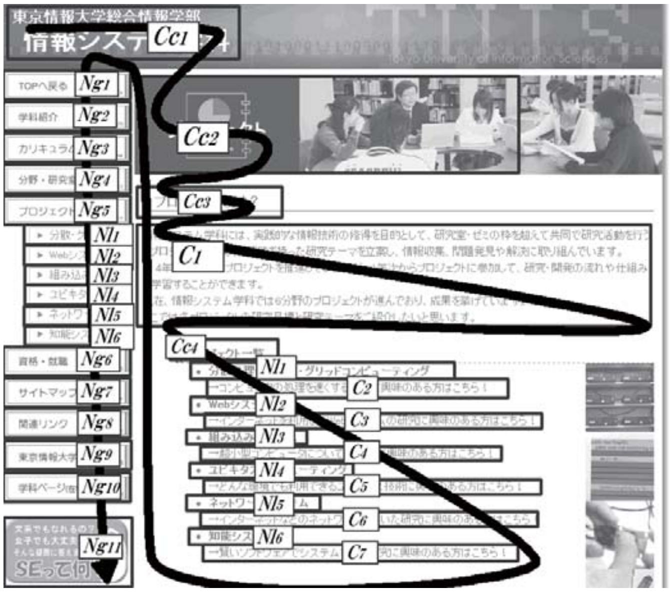

Fig. 7. Step 3 of this method

This Web page information architecture equation is shown by equation (2).

$$\begin{aligned} St \;=\; & Cc_1Cc_2Cc_3C_1Cc_4 \\ & Nl_1C_2Nl_2C_3Nl_3C_4Nl_5C_5Nl_6C_7 \\ & Ng_1Ng_2Ng_3Ng_4Ng_5 \\ & Nl_1Nl_2Nl_3Nl_4Nl_5Nl_6 \\ & Ng_6Ng_7Ng_8Ng_9Ng_{10}Ng_{11} \end{aligned} \quad (2)$$

By a design of Web site, there is the case that does not assume F-shaped reading pattern. In this case reading pattern of a user is assumed, and the flow is prescribed.

D. Step 4 : Extraction of regular expression for analysis

Arrangement of a word provided in Step 3 is abstracted only by an attribute. As a procedure, a subscript is disregarded, and arrangement of an element extracted only in the attribute of structure element is expressed with iteration (Kleene enclosure).

For example, a regular expression derived from an equation 1 is shown in equation (3).

$$St' = (Cc)^*CCc(NlC)^*(Ng)^*(Nl)^*(Ng)^* \quad (3)$$

IV. An Example

It shows an application example of this method. This example applied it on a Web page with Tokyo University of Information Science[12].

A. Step 1 & 2 : Structure elements separation and setting of variables

Fig. 8 shows the division of a structure element in this example and a variable name corresponding to each element.

B. Step 3 : Formalization

An equation to be provided from this example shows next.

$$\begin{aligned} S \;=\; & Ni_1N_1N_2N_3N_4N_5N_6N_7Cc_1N_8N_9N_{10}Si_1Si_2 \\ & Ng_1Ng_2Ng_3Ng_4Ng_5Ng_6Ng_7Ng_8 \\ & Nl_1Nl_2Nl_3Nl_4Nu_1Nu_2Nu_3Nu_4Np \\ & Cc_2Cc_3Cl_1Cl_2Cl_3\underline{Cc_4C_1N_{11}} \\ & \underline{Cc_5C_2N_{11}Cc_6C_3Nl_2}N_{14}N_{11} \\ & Cc_7Nl_5Nl_6Nl_7Nl_8Nl_9Nl_{10}Nl_{11}Nl_{12}Nl_{13} \\ & Nl_{14}Nl_{15}Nl_{16}Nl_{17}Nl_{18}Nl_{19}Nl_{20} \\ & Nt_1Nt_2Nt_3Nt_4Nt_5 \\ & Cc_8Nt_6Nt_7Nt_8Cc_9Nt_9Nt_{10}Nt_{11}Nt_{12} \\ & Ng_1Ng_2Ng_3Ng_4Ng_5Ng_6Ng_7Ng_8 \\ & Nt_{13}Nt_{14}Nt_{15}Nt_{16}C_4 \end{aligned} \quad (4)$$

C. Step 4 : Extraction of regular expression for analysis

It disregards a subscript from an equation provided in extraction Step 3. It is transformed in regular expression with iteration in this equation.

$$\begin{aligned} S' \;=\; & NiNNNNNNNCcNNNSiSi \\ & NgNgNgNgNgNgNgNg \\ & NlNlNlNlNuNuNuNuNp \\ & CcCcClClClCcCNCcCNCcCNNN \\ & CcNlNlNlNlNlNlNlNlNlNl \\ & NlNlNlNlNlNl \\ & NtNtNtNtNt \\ & CcNtNtNtCcNtNtNtNt \\ & NgNgNgNgNgNgNgNg \\ & NtNtNtNtC \end{aligned}$$

In this way it showed that it could formalize a Web page.

$$\begin{aligned} S'' \;=\; & Ni(N)^*Cc(N)^*Si(Ng)^*(Nl)^*Np(Cc)^*(Cl)^* \\ & \underline{(CcCN)^*CcCNl}(N)^*Cc(Nl)^* \\ & (Nt)^*(Cc(Nt)^*)^*(Ng)^*(Nt)^*C \end{aligned} \quad (5)$$

V. Consideration

A. Consideration of structure element lacking

It shows an analysis of the Web pages information architecture of this example that using equations provided by this method. We focus on the underline part of equation (5). It corresponds to an underline part of equation 1. As for sub equation $(CcCN)^*CcCNl$, As for the sub equation of this, $(CcCN)^*CcCNl$ is more natural than $(CcCN)^*Nl$. Therefore, it is supposed that this structure lacked N in a designing. In this case this structure should be corrected so that the flow that a user reads by the same shape is to be changed suddenly.

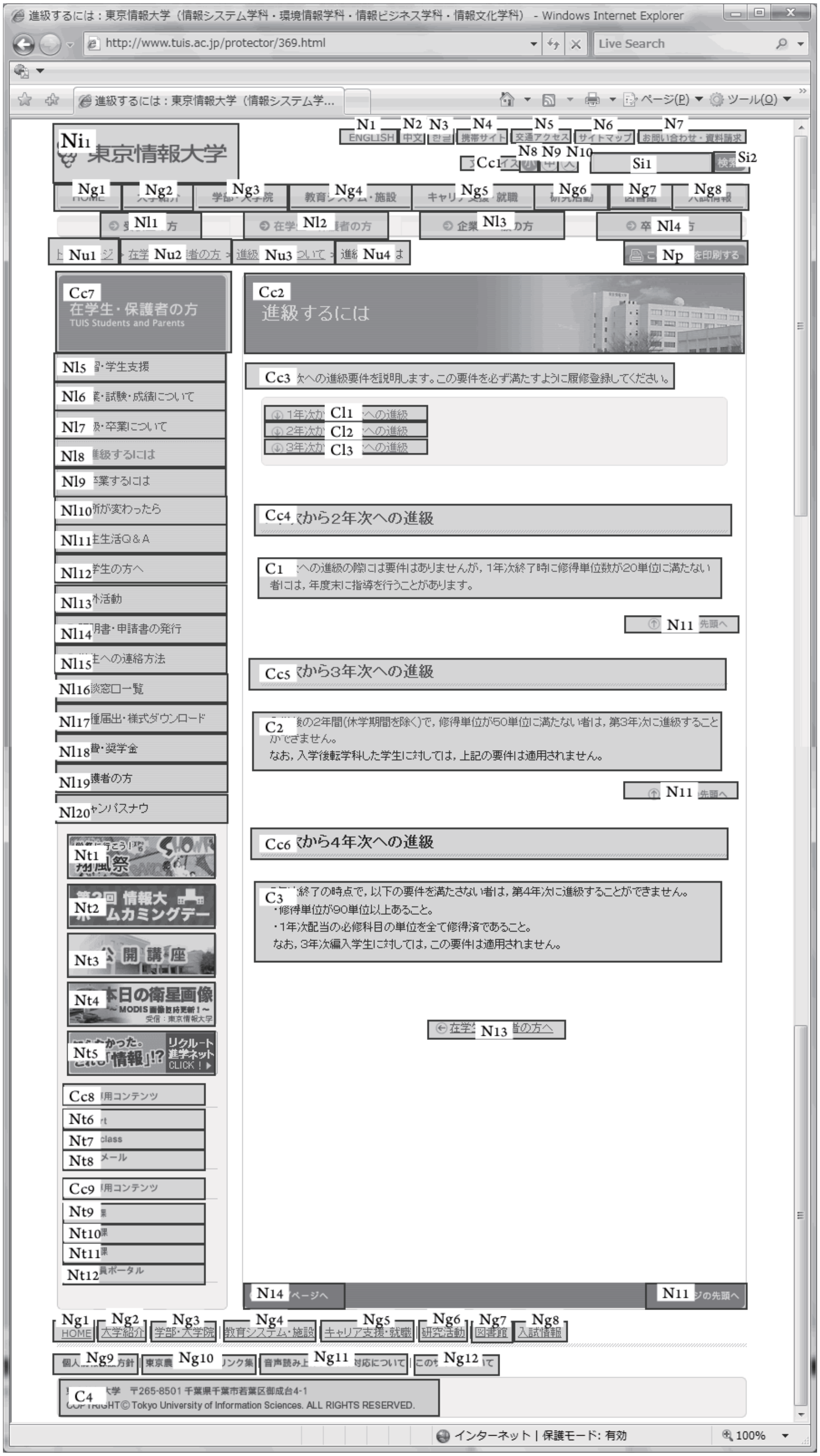

Fig. 8. Elements of example Web page

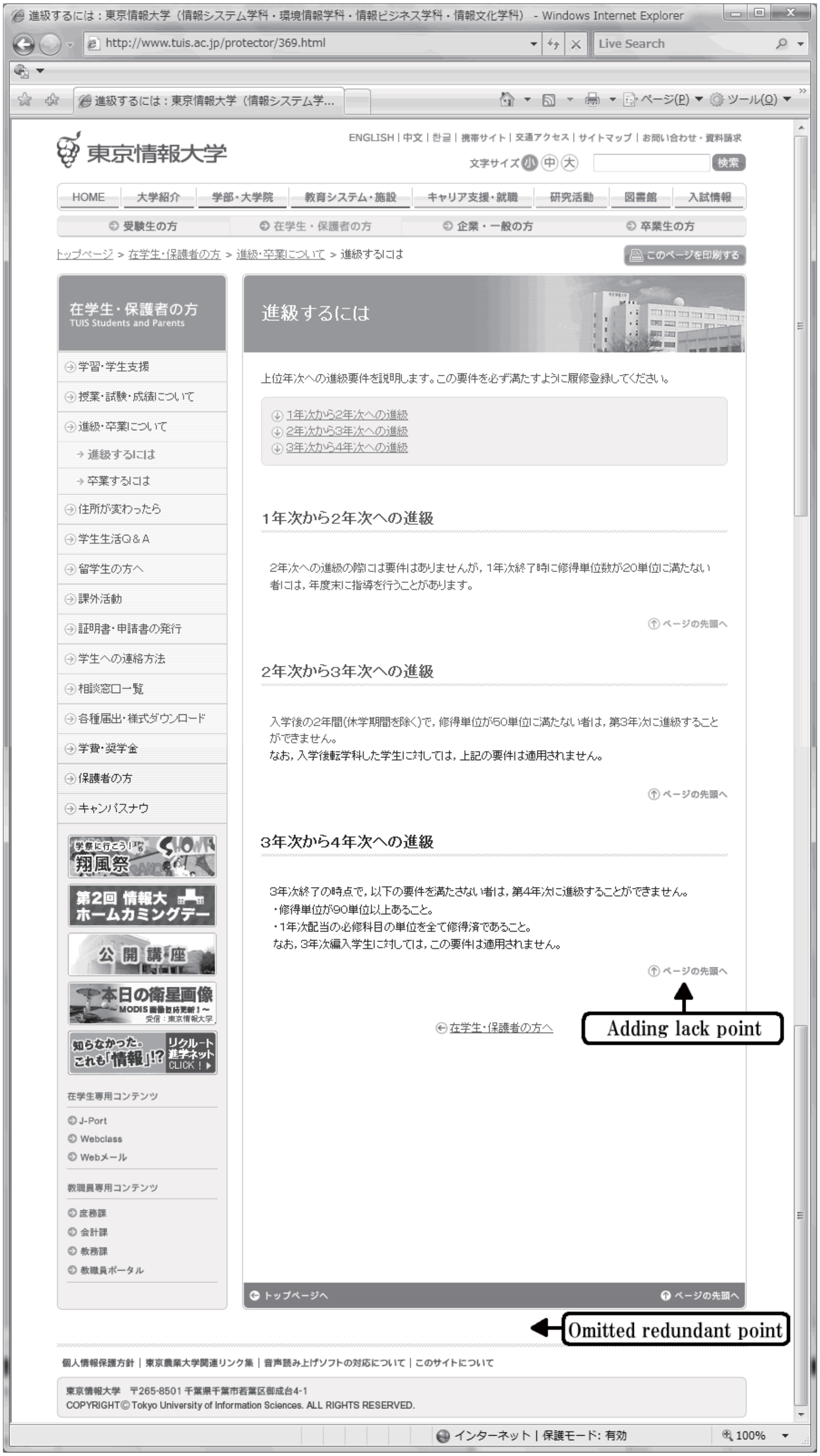

Fig. 9. Modification of this example

B. Analysis for redundant structure

There are two global navigation groups in equation (5). This both structure elements have the same links. Therefore, it is supposed that this part is redundant, and global navigations of the lower part should be cut. In addition, it is supposed that these are needless because navigation N_{11} returning to the page top just before navigation group is inserted.

C. Modification of this example

Fig. 9 shows modification of this example applicable to above considerations.

D. Notes of this method

This method can apply to only discussion. If you would apply to this method for quality analysis, you should consider to the visual design of a Web page. This method is abstract just Web information architecture that neglect colors, shapes and so on. So, this method can only focus on the one side of Web site as the information architecture. Web design has many factors as the information architecture, the chromatics, the layout design, the user interface and so on. So, Web designing must be considering these factors. This method can support one factor analysis of these. Because, considerations of this paper are not apply to modify of Web page immoderately. If you apply these considerations, you must considerable other Web design factors.

VI. CONCLUSION

This paper proposed that a method to formalize the information architecture of a Web page by regular expression. This method expresses the information architecture of a Web page by two kinds of equations formally. One is the equation that does not have iteration to keep each information structure element. Another is regular expression to have the iteration that it abstracted only in an attribute.

In addition, this paper proposed a method to analyze the information architecture of a Web page from these two equations. As a result of having applied it, this example was able to formalize structure of Web page by this method. Furthermore, by analysis with these two equations, it was able to check a redundant point and lack point about this example's information architecture.

APPENDIX A
AN ANALYSIS EXAMPLE OF PROGRAM STRUCTURE FORMALIZATION

A program structure formalized in regular expressions provides the way of analyzing the correspondences a program structure to the specifications for checking the program correctness and the traceability.

We can detect some problems including errors in a procedure which is to convert commands in a full input line into specified command lines, and which is described in pseudo codes and comments shown in Fig.10.

Although the Fig.10 contains rough specifications in the header comment part, the input command syntax is not clearly

1) An input has more than one character excluding the empty line and the lines have no character.
2) When they condition not 'more characters on the line' is true flag (such as an empty input flag) is set to 'on'.
3) A command word length is 1 to 8.

Fig. 11. Specifications for Read_Command_Line()

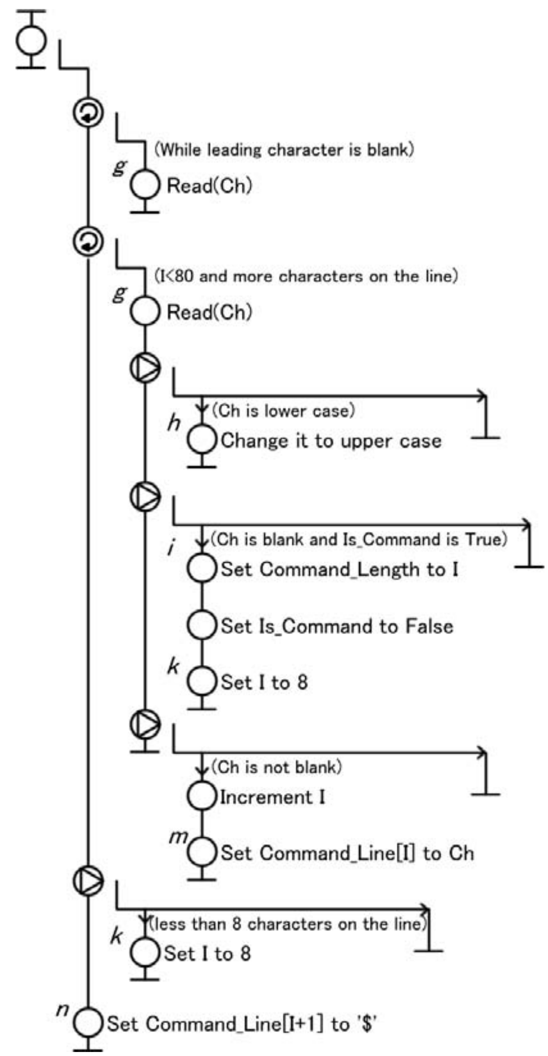

Fig. 12. Command word program in HCP

specified . It could be specified by the interpretation of codes which construct the Is_command flag mechanism in Fig.11.

The syntax of an input line can be interpreted as a character string on the alphabet $\{B, C\}$ with the following syntax, where B is a blank character, and C is any character except a blank.

$$L \equiv B^*C^+(B^+C^*)^* \tag{6}$$

Fig.12 illustrates a program in HCP diagram which is rewritten on the base of Fig.10 so as to reflects the syntax in Fig.11.

The structure in Fig.10 can be shown in the following expression by excluding mechanisms.

```
    /*Read_Command_Line(Command_Length);
    This procedure reads a full line of command, eliminates all of the blank spaces, converts
    lower case letters into upper case letters, and stores the command line in array
    Command_Line. It also identifies the end of command line by placing a "$" as the last
    character in array Command_Line. The argument Command_Length will return a 1 if the
    command is in short form. If the command is in long form, this argument returns the
    length of the command. A value other than 1 means the command is in long form.
    Regardless of the length of the command entered (short or long), the first 8 positions
    in array Command_Line are set aside to store the command word.
    */

    Local Declarations
      Define I as integer
      Define Ch as character
      Define Is_Command as Boolean
    Begin Read_Command_Line
      Set I to 0 /*character count*/
      Set Is_Command to True
      Set Command_Line to a Black line
      While leading character is blank
g       Read(Ch)
      End-while
      While I < 80 and more characters on the line
g       Read(Ch);
        If Ch is lower case then
h         Change it to upper case
        End-if
        If Ch is blank and Is_Command is True then
i         Set Command_Length to I
          Set Is_Command to False
k         Set I to 8 /*the longest command*/
        End-if
        If Ch is not blank then
          Increment I
m         Set Command_Line[I] to Ch;
        End-if
      End-while
      If less than 8 characters on the line then
k       Set I to 8
      End-if
n     Set Command_Line[I+1] to '$'
      Finish the current line
    End Read_Com_Line
```

Fig. 10. Command word program

$$P \equiv g^*(g(h+\varepsilon)(ik+\varepsilon)(m+\varepsilon))^*(k+\varepsilon)n \tag{7}$$

,where the variable g,h,i,k,m,n are illustrated in the Fig.10.

A. Command head character problem

There are two problems in the pseudo code of Fig.10. The first is that the head character except a blank is skipped to end. The term g^* in the expression (7) could not be coded, if the condition in the pseudo code 'while leading character is blank' is written as ' Ch==" " ' for the propositions of the iteration, since imperative construct g has the context over its iteration. The condition of the iteration must be placed after g, because character testing can be done only after getting a character. That is the logic of this program should be g^+ in order to check after to read a character($= g$).

We must assume 'Set Ch to blank' as initialize value of Ch, or change the iteration type to a post-tested one. Here g^* in the expression (7) is changed to g^+. Therefore, the structure P should be changed into the structure P'.

To clear the discussion it introduced a variable α, $\alpha \equiv (h+\varepsilon)(ik+\varepsilon)(m+\varepsilon)$.

$$P' \quad \equiv \quad g^+(g\alpha)^*(k+\varepsilon)n \tag{8}$$

The second is that the lack of a head character of a command word from observing g in the expression (8). The term $g^+(g(\alpha))^*$ means to overwrite a head character in output with the second character in input. The structure P' should be modified to correct the error as follows;

$$P'' \quad \equiv \quad g^+(\alpha g)^*\alpha(k+\varepsilon)n$$

B. Placement of ' \$ ' problem

We focus on the ' \$ ' placement process.

On complex structures it is effective to highlight specific constructs by replacing don't-care constructs with ε The variables k,m,n are highlighted in program structure P'' because the constructs which are composed of the process can be identified by detecting ones which are concerned with variable I in pseudo code. The following program structure is introduced.

$$(km)^*(\varepsilon + k)n = ((km)^* + (km)^*k)n$$

,where the variable k means 'set I to 8', the variable m means 'increment I' and the variable n means 'output I'.

It should be omitted k from $(km)^*$ as redundant and erroneous process because the maximum value of variable I is 8 it has shown by specifications. About k it should be omitted from $(km)^*k$ in a similar way.

Therefore, $(ik + \varepsilon)$ in α must be modified to $(i + \varepsilon)$.

REFERENCES

[1] Peter Morville, Louis Rosenfeld, "Information Architecture for the World Wide Web", Oreilly & Associates Inc, Nov 27, 2006
[2] Satou Tadamasa, "A Formalization by S-Algebra for Program Algorithm Described in HCP-Charting,",IPSJ Vol.27 No.6 ,1986 (in Japanese)
[3] Satou Tadamasa, "Universal Form for Program Iteration Structure", ICCIT2000 Proceedings PP.240-243 ,2001
[4] Satou Tadamasa, Kishimoto Yorinori, "An Analysis Method of Program Traceablity to the Specification through Program Construct Formalization", Advances in Systems Engineering, Signal Processing and Communications, pp19-24,2002
[5] Satou Tadamasa, Kishimoto Yorinori, Md. Atiqul Islam, "Program Structure Formalization and Traceability Analysis with Examples", Proceedings of 5th International Conference on Computer and Information Technology , pp537-542 ,2002
[6] Satou Tadamasa, Kishimoto Yorinori, "Program Code Analysis Focused on its Structure", WSEAS TRANSACTION on COMPUTERS, Issue 1, Volume 2, pp24-29
[7] Yorinori Kishimoto, Takako Itou, Tadamasa Satou, "Program Structure Formalizing Technology for Static Analysis", WSEAS TRANSACTIONS on SYSTEMS,Issue 4, Volume 3, pp1691-1698
[8] Kishimoto Yorinori, Satou Tadamasa, "Formalizing Effects of Program Structure", The 2004 International Conference on Software Engineering Research and Practice, pp.505-509 ,2004
[9] Jakob Nielsen's Alertbox, "F-Shaped Pattern For Reading Web Content", http://www.useit.com/alertbox/reading_pattern.html, April 17, 2006
[10] Tokyo University of Information Sciences Department of Information Systems Web Site, http://www.infosys.tuis.ac.jp/
[11] Google.co.jp, http://www.google.co.jp/
[12] Tokyo University of Information Sciences Web Site, http://www.tuis.ac.jp/
[13] M.A.Jackson, "Principles of Program Design", Academic Press, 1975
[14] J.W.Hughes, "A Formalization and Explication of the Michael Jackson Method of Program Design", SOFTWARE-PRACTICE AND EXPRENCE Vol.9 PP.192-201,1979
[15] Itou Takako, Kishimoto Yorinori, Satou Tadamasa, "A Marking Method for Programming Skills Based on Program Structural Analysis", 6th INTERNATIONAL CONFERENCE ON COMPUTER AND INFORMATION TECHNOLOGY, pp.804-807 ,2003
[16] Satou Tadamasa, Kishimoto Yorinori, Hamana Takahiro, "A Variety Analysis Method for Programs Written in a Training Course", 7th INTRNATIONAL CONFERENCE ON COMPUTER AND INFORMATION TECHNOLOGY, pp.553-558 ,2004
[17] Yorinori KISHIMOTO "A Traceability Analysis Method for Event Driven System", 2006 WSEAS International Conference on APPLIED COMPUTER SCIENCE ,2006
[18] Yorinori Kishimoto, "A Traceability Analysis and its Improvement for Event Driven System," WSEAS TRANSACTIONS on INFORMATION SCIENCE and APPLICATIONS,Issue 4, Volume 4, pp843-849, 2007
[19] Yorinori Kishimoto, "An Idea of Formalizing Web page Information Architecture", 2008 WSEAS International Conference ACS'08, Nov 2008

Yorinori KISHIMOTO received the M.S. degree from Shimane University, Shimane, Japan in 2001, respectively.

And he received the Ph.D. degree from Nagoya Institute of Technology, Aichi, Japan in 2005.

After he was working in Graduate School of Kobe Institute of Computing from 2005 to 2007, he moved to Tokyo University of Information Sciences in 2007. His current research interest is Web information architecture and quality of information systems.

He is a lecturer in Department of Information Systems of Tokyo University of Information Sciences.