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.

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 \( \cdot \) (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 question-
naire 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 adja-
cent. 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 archi-
tecture’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 architec-
ture elements as a purpose to usefulness measurement of a
Web page.

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 thersaurus. 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 consid-
ered 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.
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).

$$X = Cc_1 Nl_1 Nl_2 Nl_3 C_1 Ng_1 Ng_2 Ng_3$$

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

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.

\[
A = C_1C_2C_3N_1N_2N_3N_4
\]
\[
A' = (CcCN)^*CcCNl
\]

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.

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 \(N_{l1}-N_{l6}\) 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.
This Web page information architecture equation is shown by equation (2).

\[ S_t = C_1C_2C_3C_4 \\
N_1C_2N_2C_3N_3C_4N_4C_5N_5C_6C_7 \\
N_9G_2N_3G_4N_5G_6G_7G_8 \\
N_1C_8N_1C_9N_2C_3N_3N_4N_5N_6N_7C \\
N_9G_2N_3G_4N_5G_6G_7G_8 (2) \]

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).

\[ S_t' = (C_c)^\ast CC_c(N_c)^\ast(N_g)^\ast(N_l)^\ast(N_g)^\ast \quad (3) \]

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.

\[ S = N_1N_2N_3N_4N_5N_6N_7C_1N_8N_9N_{10}N_{11}S_{12} \\
N_9G_2N_3G_4N_5G_6G_7G_8 \\
N_1N_2N_3N_4N_5N_6N_7N_{12}N_{13} \\
N_9G_2N_3G_4N_5G_6G_7G_8 \]

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.

\[ S_t' = N_1N_2N_3N_4N_5N_6N_7N_{12}N_{13}C_1N_2N_3N_4N_5N_6N_7N_{12}N_{13} \\
N_9G_2N_3G_4N_5G_6G_7G_8 \]

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

\[ S_t'' = N_1(N_c)^\astCc(N_c)^\astSi(N_g)^\ast(N_l)^\astN_pCc(C_l)^\ast(N_l)^\ast(N_g)^\ast(N_l)^\astC \quad (5) \]

V. CONSIDERATION

A. Consideration of structure element lacking

It shows an analysis of the Web page's 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 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.

\[
L \equiv B^+C^+ (B^+C^+)^+ \quad (6)
\]

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.
Fig. 10. Command word program

\[ P \equiv g^+ (g(h + \varepsilon)(ik + \varepsilon)(m + \varepsilon))^{*} (k + \varepsilon)n \]  \hspace{1cm} (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 \( \alpha \), \( \alpha \equiv (h + \varepsilon)(ik + \varepsilon)(m + \varepsilon) \).

\[ P' \equiv g^+ (g\alpha)^* (k + \varepsilon)n \]  \hspace{1cm} (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;

\[ P'' \equiv g^+ (\alpha g)^* \alpha (k + \varepsilon)n \]

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 \( \varepsilon \) 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 \( \alpha \) must be modified to \((i + \varepsilon)\).

REFERENCES

[16] Satou Tadamasa, Kishimoto Yorinori, Hamana Takahiro, "A Variety Analysis Method for Programs Written in a Training Course", 7th INTERNATIONAL CONFERENCE ON COMPUTER AND INFORMATION TECHNOLOGY, pp.553-558, 2004