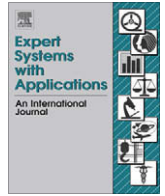




Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Enhancing the precision of content analysis in content adaptation using entropy-based fuzzy reasoning

Rick C.S. Chen^a, Stephen J.H. Yang^{a,*}, Jia Zhang^b

^a Department of Computer Science and Information Engineering, National Central University, Taiwan

^b Department of Computer Science, Northern Illinois University, USA

ARTICLE INFO

Keywords:

Content adaptation
Semantic coherence
Longest common subsequence (LCS)
Fuzzy reasoning
Entropy
Inductive reasoning

ABSTRACT

Content adaptation is a well-known technique to help portable devices present Web pages as smoothly as desktops do. Because of limited I/O and weak transmission capability, adaptations are usually performed by either transcoding or resizing multimedia components. In this paper, we propose a novel semantic coherence-retained content adaptation approach, namely functionality sense-based content adaptation (FSCA). Our goal is to avoid semantic distortions when rearranging a Web page on different screen sizes. Simulating entropy-based fuzzy reasoning in human cognition, we introduce *Relevance of Functionality* (RoF) to quantitatively represent the similarity intensity between two presentation objects (groups) based on their functionalities. We present an algorithm of calculating RoF and a procedure that uses RoF to decide content adaptation degree. Our experiments verify the feasibility and effectiveness of FSCA.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

In recent years, a number of portable devices have emerged to enable people to access information easily. Users can surf the Internet via PDAs and smart phones at anytime from anywhere. However, most Web pages are originally designed for display on desktops rather than on portable devices with much smaller screens (e.g., people may have to repeatedly scroll in the middle of browsing on a phone for a comprehensive view). Limited computing power and bandwidth also restrain portable devices from some potential applications such as real-time video. To support smooth display on portable devices, many studies have developed various content adaptation techniques (CA for short) that focus on component resizing and layout rearrangement. Here components, or presentation components, refer to various multimedia objects (objects for short, such as text, video, audio and flash) and HTML constructs (such as *table* and *div* in a Web page). Component resizing aims to enhance component readability by adjusting their sizes; layout rearrangement aims to change a display layout from multi-column to single-column to simplify user operations on small screens. Based to these two methods, the adapted Web page shown in Fig. 1(a) provides better user understanding and requires only vertical scrolling, in contrast to its original Web page shown in Fig. 1(b) that requires repeated horizontal and vertical scrolling activities.

However, the two methods do not consider semantic coherence between presentation components. As a result, they may lead to

semantic distortions including correlation distortions and integrity distortions. A correlation distortion means that a content adaptation loses some correlation relationships implied in the original Web page. A correlation refers to a type of semantic relations between presentation objects. For example, a correlation exists between a title and its corresponding article. Fig. 2(a) illustrates a correlation distortion. The upper part shows a table of an original Web page, which comprises a header row (marked as “T” object) and a list of content rows (marked as “N” object). The correlation between the two objects implies that each row in the “N” object follows the format specification defined in the “T” object. This means that each content row contains two parts, with *date* on the left and *event* on the right. The lower part of Fig. 2(a) shows the resulted display on a phone after content adaptation. To fit into the small screen, the component resizing method shortens the width of Object “T”; the layout rearrangement method breaks every row in the Object “N” into two rows. The resulted “Object N” does not follow the specification in “Object T” any longer. The correlation between the two objects is lost.

An integrity distortion means that the integrity of a composite object is lost by a content adaptation. A composite object refers to a presentation object that is composed of at least two individual objects with a specific order. As shown in the upper part of Fig. 2(b), the Harvard university seal is a composite object that is composed of six individual figures in the visual order. The lower part of Fig. 2(b) shows that its integrity is lost because of a column-wise adaptation (multiple-to-single column adaptation).

To avoid semantic distortions, analyzing semantic coherence between presentation objects in a Web page is an essential step in content adaptation techniques. Our previous study proposes

* Corresponding author. Tel.: +886 3 422 7151x57102.
E-mail address: jhyang@csie.ncu.edu.tw (S.J.H. Yang).

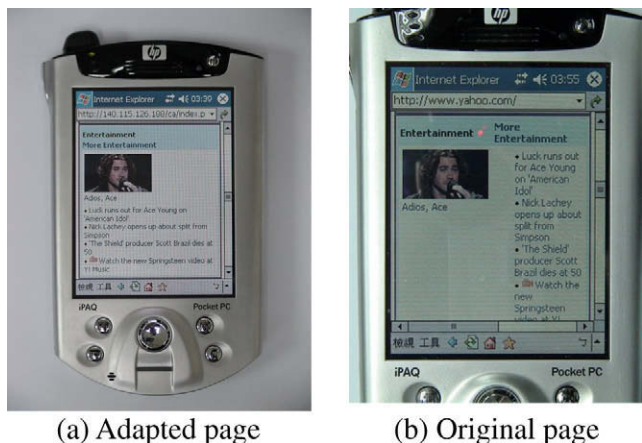


Fig. 1. Comparison of a Web page with and without content adaptation.

an algorithm to identify Units of information (UOIs) that represent presentation groups with high semantic coherence between comprising presentation objects (Yang, Zhang, Chen, & Shao, 2007). Since our algorithm is based on predefined rules, it may not be flexible and precise enough to identify some UOIs in Web pages containing vague information. Such vague semantic information, however, can be easily recognized by human beings with fuzzy sense. For example, people can determine whether two presentation objects play similar roles (similar functionalities) by observing their features and locations. In general, two objects playing the same roles look similar on features and are usually physically close to each other. Fig. 3 shows a part of the official site of the baseball league. Four presentation objects are identified: A is a table containing the title, picture, and abstract of the top news; B is a table of more headlines; C and D are two headline titles as two rows in table B. As shown in Fig. 3, C and D have similar functionalities (they show two headlines and they aim to guide readers), so that they show similar presentation styles and they are next to each other in the two rows in the same table. Meanwhile, A and B have different functionalities and they do look different. In short, the intensity of semantic coherence between C and D is higher than that between A and B. Hence, semantic distortions have more chances to occur between C and D.

In this research, we simulate human sense on fuzzy reasoning to establish a functionality sense-based content adaptation (FSCA) mechanism protecting semantic coherences at the time of content



Fig. 3. Example of a part of the official site of major league baseball.

adaptation. To realize this goal, we introduce *Relevance of Functionality* (RoF) to quantitatively represent the similarity intensity between two presentation objects (groups) based on their functionalities. We propose an algorithm that automatically calculates RoF and a procedure that uses RoF to decide content adaptation degree. Our experiments show that our mechanism resulted in a significant drop of semantic distortion rate (a fifty-percent drop from 16% to 8%) and the adapted results received a high acceptance rate from participating students.

The remainder of this paper is organized as follows. Section 2 reviews related studies. In Section 3, we present our similarity quantification algorithm, SimilarityToRoF translation algorithm, and our content adaptation algorithm. Section 4 shows the architecture of our FSCA engine. Section 5 presents our experiments and findings. Section 6 makes conclusions.

2. Related work

The conventional approach of providing Web content supporting mobile devices is to prepare specific versions (formats) of the same content for mobile devices. For example, a Web page holds one HTML version supporting desktop devices and one Wireless Markup Language (WML) version supporting wireless devices. This approach is straightforward but labor-intensive yet inflexible. Content providers have to prepare different layouts and formats for the same Web content, which results in tremendous overhead. Even worse, any change in the content may result in consequent changes in every related version, which is highly inflexible and may easily cause inconsistency. Considering Web content typically

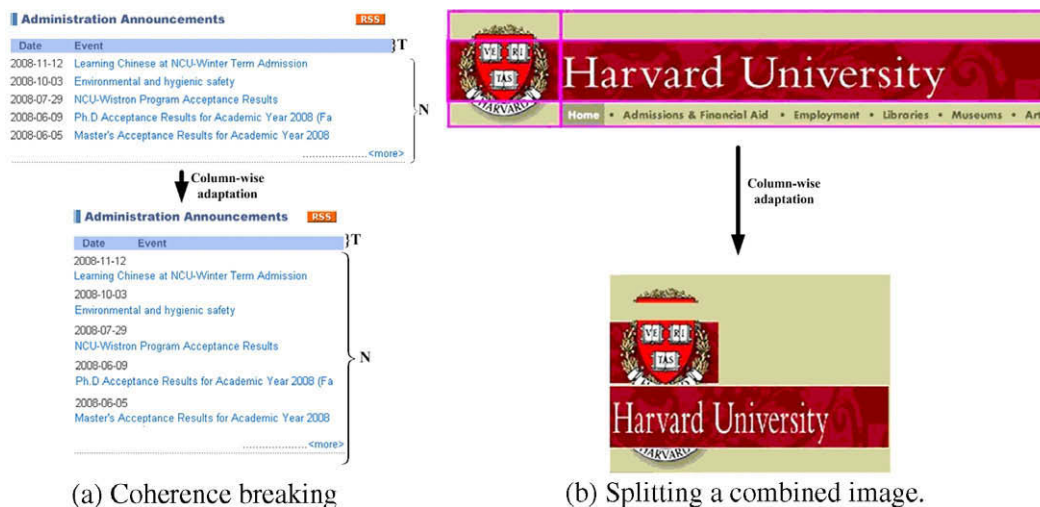


Fig. 2. Examples of semantic distortion.

undergoes frequent changes, this traditional approach is neither practical nor feasible for content delivery.

Content adaptation techniques thus are developed to dynamically transform Web pages into appropriate formats, such as Google, Yahoo. Such techniques may optimize the content delivery service for different devices. For example, most PC users have a screen resolution of at least 1024×768 pixels, while the displays on a Personal Digital Assistant (PDA) maxes out at 240times320 pixels, yet a mobile phone screen possesses 176times208 pixels or less. In addition, other Web access conditions for wireless devices may vary such as network connectivity, processing power, storage, display, and format handling capabilities. To date, many multi-channel or content publishing tools have appeared to support content adaptation facilities. Typical examples are Oracle application server wireless (Oracle and Application Server Wireless), Sun Java system portal server mobile access (Sun and System Portal Server Mobile Access), and WebSphere transcoding publisher (IBM).

However, these publishing tools require that the adaptable content be developed using the same tools and platforms from the beginning. Requiring that all Web content be developed in a formalized way, maybe in a rather long period of time, is neither feasible nor even desirable. Meanwhile, there are still many existing and appearing Web content originally designed and developed to be displayed on desktops using HTML. Requiring that all HTML pages be regenerated to support multiple channels is even more impractical. How to make these large-screen-oriented HTML pages become automatically and transparently adaptable and accessible to mobile users remains greatly necessary yet highly challenging.

Due to this reason, researchers have developed many HTML-oriented content adaptation prototypes in the recent years. Among them, Phan, Zorpas, and Bagrodia (2002) propose a middleware, called Content Adaptation Pipeline (CAP), to perform content adaptation on any complex data types in addition to text and graphic images. They use eXtensible Markup Language (XML) to describe all elements in a content hierarchy. Berhe, Brunie, and Pierson (2004) present a service-based content adaptation framework. An adaptation operator is introduced as an abstraction of various transformation operations such as compression, decompression, scaling, and conversion. Burzagli, Emiliani, and Gabbanini (2009) discuss the aspects related to Design for All (D4All) that is a developer-driven concept to build services of various device types. XML-based adaptation is the major example used to illustrate their concept. Lemlouma and Layaida (2004) propose an adaptation framework, which defines an adaptation strategy as a set of description models, communication protocols, and negotiation and adaptation methods. Lee, Chandranmenon, and Miller (2003) develop a middleware-based content adaptation server providing transcoding utilities named GAMMAR. A table-driven architecture is adopted to manage transcoding services located across a cluster of network computers. These works show some proofs-of-concept of Web content adaptation. However, the actual implementation is still in a primary phase. How to map from constraints to adaptation operators is still unsolved. The scalability issue is a bottle neck as well.

Some researchers focus on content decomposition methods. Chen et al. (2002) propose a block-based content decomposition method for quantifying content representation. An HTML page is factorized into blocks, each being assigned a score denoting its significance. This method enables content layout to become adjustable according to the region of interest, attention value, and minimum perceptible size. Ramaswamy, Iyengar, Liu, and Douglis (2005) propose an efficient fragment generation and caching method based on detection of three features: shared behavior, lifetime, and personalization characteristic. The smallest adjustable element in these two approaches is a composite of objects (i.e., text, image, audio, and video). This granularity of decomposition is too

large for mobile device screens; therefore, they are not suitable for mobile content adaptation. Zhang, Zhang, Quek, and Chung (2005) propose extensions to CC/PP to enable transformation descriptions between various receiving devices. However, their work is based on the fact that the presented content already possesses multiple presentation versions.

MobiDNA (Hua, Xie, Liu, Lu, & Ma, 2006) proposes an adaptation algorithm to improve readability of Web content and enhance the caching strategy for reducing browsing latency. Its adaptation process adjusts the size of Web content according to semantic blocks, which are defined as continuous content units that do not include two or more fragments within their content scopes. However, this approach must be implemented on client side.

XAdapter (He, Gao, Hao, Yen, & Bastani, 2007) is another extensible content adaptation system, where Web objects are classified into objects (structure, content, and pointer objects) and adaptation techniques for structure objects (e.g., HTML tables). Nevertheless, the visual coherence between objects may be broken, because the approach does not consider whether some layouts cannot be rearranged.

In our previous research (Yang & Shao, 2006; Yang, Zhang, & Chen, 2008), we present a JESS-enabled context elicitation system featuring an ontology-based context model to formally describe and acquire contextual information of receiving environments. In Yang et al. (2007), we present a UOI-based content adaptation method, which automatically detects semantic relationships among comprising components in Web contents, and then reorganizes page layout to fit handheld devices based on identified UOIs. In Yang, Zhang, and Chen (2007), we present an ontology-based context model supported by context query and phased acquisition techniques. However, our experiments revealed that this syntax-oriented detection may not always lead to satisfactory results. In this paper, we apply fuzzy reasoning techniques to propose a functionality sense-based content adaptation (FSCA) mechanism, which quantifies similarity between presentation projects and uses it to enhance content adaptation.

Table 1 compares our proposed approach (FSCA) with related methods (XAdapter, UOI, and MobiDNA) according to a list of perspectives. The first two perspectives (layout rearranging and thumbnail) focus on adaptation process. Each method has its own adaptation process, layout rearrangement or thumbnail presentation, except SSI. The next two perspectives (coherence detection and blur detection) examine content analysis. Most approaches perform well in maintaining the coherence of content, except Xadapter that is poor in coherence detection although it can prevent blurring caused by shrinking texts or images. Regarding system architecture, our method and Xadapter are proxy-based approaches featuring flexibility. CCAP and SSI use an open-source software, Tidy, to fix invalid HTML tags and give source code a well-formed layout. However, parsing source code may have an impact on performance. Not every system requires Tidy to fix source code due to different system designs. Xadapter and SSI have a capability of analyzing Cascading Style Sheets (CSS) (W3C) content, which function can improve the precision of Web content analysis.

Table 1
Comparison among various approaches.

	FSCA	XAdapter	UOI	MobiDNA
Layout rearranging	✓	✓	✓	
Thumbnail				✓
Coherence detection	✓		✓	✓
Blur detection		✓		
Proxy-based	✓	✓		
Tidy (XHTML)			✓	
CSS analysis	✓	✓		Unknown
Javascript		Unknown		✓

In this research, we simulate human sense on fuzzy reasoning (Ross, 2004) to detect semantic coherences between presentation objects. One critical step of fuzzy reasoning is to build the membership function that decides whether a member belongs to a fuzzy set. Ross (2004) introduces the inductive reasoning that applies the information entropy to measure the uncertainty based on composition proportion of a collected training data set. Tsai, Cheng, and Chang (2006) also propose the Entropy Principle Algorithm to build the membership function of fuzzy sets of linguistic variables based on the inductive reasoning. Applying the concept of fuzzy logic to content adaptation, we use the similarity between presentation objects to help explore entropy.

3. FSCA method and solutions

Users can use formats (e.g., table and div) and attributes (e.g., color and font) to decide the semantic relations among presentation objects groups in a Web page. Formats and attributes affect the features and functionalities of presentation objects. If a pair of presentation groups in a Web page have been presented in similar features and structures, they are likely to be considered as semantically relevant. Therefore, one way to simulate the human sense is to quantify the similarity of two presentation groups.

3.1. Define the scope of comparison units

To explore how to calculate the similarity between two presentation groups, we first have to define the scope of a presentation group for comparison purpose. An intuitive idea is to take every presentation object as a comparison unit, e.g., an icon and a text area. However, not every presentation object is significant on a specific Web page. Some objects, such as the symbols x and y shown in Fig. 4, play a role of decorating other presentation objects. If they are all considered as comparison units, the comparing complexity may be increased significantly without adding many values.

Thus, we define a comparison unit as a minimum semantic group (SG) containing semantic meanings, when it stands alone. Based on our observations, we consider three types of semantic groups: a row of a table, a table and a division (<div>). These types each may contain multiple presentation objects. By identifying every semantic group and their relationships from its HTML syntax, a Web page can be translated into a structure tree. Fig. 4 shows

our idea of a comparison unit and how to build a structure tree from an HTML page. Fig. 4(A) shows the original Web page of the NCU, with semantic groups manually identified and marked (For illustration purpose, the original HTML code is not listed). Fig. 4(B) shows how the identified semantic groups are organized into a structure tree. The direct children under a node are arranged from left to right according to their adjacent positional relations in the original HTML code. For example, $TR_{2,1}$ and $TR_{2,2}$ represent two top adjacent rows in $Table_2$, so they show as the first and second child nodes under the node $Table_2$. Note that this structure tree comprises two types of SGs: *Table* stands for a table and *TR* stands for a row of a table.

3.2. LCS-based similarity calculation

Ideally, similarity comparison shall be conducted between any pair of semantic groups. Such a comparison between a pair of SGs requires a time complexity of $O(n^2)$, where n is the average number of features in a SG. Assume that the number of SGs in a Web page is m . The complexity of calculating all the similarities between its contained SGs is $O(m!n^2)$. As a matter of fact, we found that it is unnecessary to consider all random pairs. A pair of SGs having strong semantic coherences typically shows adjacent with each other visually. For example, $TR_{2,1}$ and $TR_{2,2}$ in Fig. 4 have the similar functionality of being a top-level menu item of the page. They thus have high similarity and they are next to each other on the page. Based on this consideration, in this project, we focus on performing similarity comparison between every pair of adjacent SGs (PASG for short). Such a strategy could reduce the complexity to $O(mn^2)$. In general, we can consider n as a constant, comparing to the variable value of m in any Web page.

After carefully studying the semantic groups with similar functionalities, we found that they usually display similar visual features on the screen, and consequently comprise similar attributes in the corresponding HTML files. For example, SGX (Row 1) and SGY (Row 2) in Fig. 5 both act as a headline link. They look similar in the display (except with different content) and their HTML code show nine common features (detailed descriptions will follow). From our observations, we thus propose a hypothesis: the similarity of a PASG can be measured by the number of common features in their HTML code.

Apparently, the order of attributes in the HTML code has to be taken into consideration. As shown in Fig. 5, if the two attributes

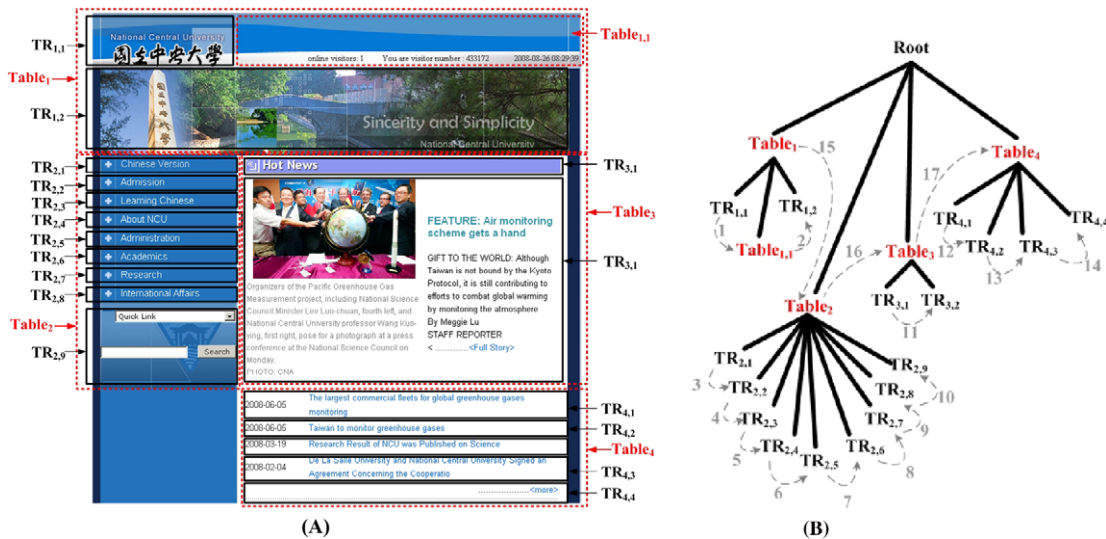


Fig. 4. Construction of structure tree for NCU website.

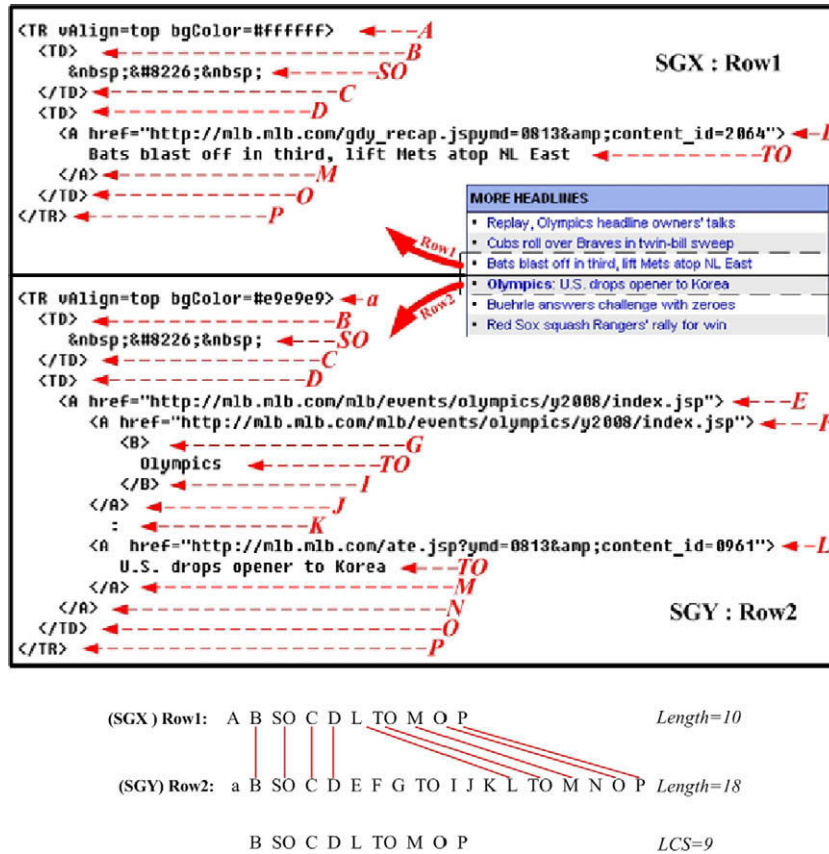


Fig. 5. An example of LCS calculation between semantic groups

(BC) in SGY change their order into (CB), they should not be considered as common attributes with SGX. Thus, we refine our hypothesis as follows: the similarity of a PASG can be measured by the maximum number of feature subsequences common to both SGs in their HTML code.

Based on our hypothesis, we formalize the question of calculating the similarity of a PASG as a Longest Common Subsequence (LCS) problem. Given two arbitrary sequences X and Y, where X contains m elements and Y contains n elements ($X = \{x_i\}_{i=1}^m$, $Y = \{y_j\}_{j=1}^n$), LCS is to find out the longest subsequences (elements) common to both X and Y:

$$LCS[X, Y] = \max\{Z_k | 1 \leq k \leq p, 0 \leq p \leq \min(m, n)\} \Rightarrow$$

$$\{Z_k\}_{k=1}^p = \{X_{z'_k}\}_{k=1}^p, Z'_k = k + l, 0 \leq l \leq m - k, \forall i, j \in [1, p], i < j, Z'_i < Z'_j \wedge$$

$$\{Z_k\}_{k=1}^p = \{Y_{z''_k}\}_{k=1}^p, Z''_k = k + l', 0 \leq l' \leq n - k, \forall i, j \in [1, p], i < j, Z''_i < Z''_j$$

We can model the SG similarity comparison problem as a LCS problem in the following way:

1. Each SG represents a sequence in LCS problem;
2. Each tag in a SG is identified as an element in the sequence and assigned an ID;
3. The objective is to match all the same elements between the two sequences in a strictly increasing order, and find the longest common element subsequences.

Fig. 5 illustrates the formalized LCS problem between SGX and SGY. 10 elements are identified in SGX and 18 in SGY. For illustration purpose, Fig. 5 uses the same IDs to represent matched elements, e.g., B, C, and D. Fig. 5 is presented as:

$$SGX = \langle A, B, SO, C, D, R, TO, M, O, P \rangle \quad \text{and,}$$

$$SGY = \langle a, B, SO, C, D, E, F, G, TO, I, J, K, R, TO, M, N, O, P \rangle$$

Now that we have formed an LCS problem, the challenge is how to decide whether two elements are identical or not. Different than the LCS problem in bioinformatics research where each element is a character, an element in our LCS problem is a tag that may comprise multiple (attribute, value) pairs (e.g., elements A and L in SGX in Fig. 5).

As shown in Fig. 5, different types of HTML tags are considered as different elements, e.g., B and C. Note that the same types of tags with different attribute values may cause different presentation results. For example, elements A in SGX and a in SGY share the same tag <TR>. But the values for their attribute "bgcolor" are different: bgcolor #ffffff in A and #e9e9e9 in a. This difference represents two different colors and obviously will lead to different presentations. Therefore, these two tags (A and a) are considered as different elements. On the other hand, if comprising different attributes values do not affect presentations, the two tags should be considered identical. Take the two L elements in SGX and SGY in Fig. 5 as an example. Although containing different values for their hyperlink attributes (two different link addresses), they should be considered as the same element (identical as labeled by L) because the hyperlink attributes will not affect their presentations. In summary, we only need to consider different attributes values if they will affect the presentation of the SGs (e.g., color).

The types of comprising presentation objects may directly make two SGs look different. For an instance, a figure looks different from text. Hence, if a SG contains presentation objects, their types are the only factor we concern during comparisons. In this project, without losing generality, we consider seven types of presentation objects: (1) symbol object (SO) for symbols, (2) text object (TO) for

text, (3) *still image object* (SIO) for jpg, bmp and tiff, (4) *video object* (VO) for avi, wmv, and mpg, (5) *dynamic image object* (DIO) for png and gif, (6) *flash object* (FO) for swf, and (7) *audio object* (AO) for mp3 and wav.

Using the above rules, it is easy to see the longest common sequence in Fig. 5 is $\langle B, SO, C, D, R, TO, M, O, P \rangle$ and its length is 9.

To automate the process of finding the longest common sequence in two SGs, we apply the dynamic programming (Cormen, Leiserson, Rivest, & Stein, 2001) that is an algorithmic technique commonly used in sequence analysis. Given a pair of adjacent SGs, $X = \{x_i\}_{i=1}^m$, $Y = \{y_j\}_{j=1}^n$. A prefix denotes a contiguous subsequence in a SG from the beginning. Let $|LCS[i, j]|$ be the set of LCS of the prefixes $X[i] = \{x_a\}_{a=1}^i$ and $Y[j] = \{y_b\}_{b=1}^j$. Recall that our goal is to find out the similarity between two SGs, we are interested in the length of the longest common subsequence that can be calculated using a recursive formula as follows:

$$|LCS[i, j]| = \begin{cases} 0 & \text{if } (i = 0 \text{ or } j = 0) \\ |LCS[i - 1, j - 1]| + 1 & \text{if } (i, j > 0 \text{ and } x_i = y_j) \\ \max(|LCS[i, j - 1]|, |LCS[i - 1, j]|) & \text{if } (i, j > 0 \text{ and } x_i \neq y_j) \end{cases}$$

The length of the longest common elements used in the PASG is the value of $|LCS[m, n]|$ that is the length of the LCS of SGX and SGY. However, since the lengths of SGs are inconsistent, the length of LCS alone may not accurately reflect the similarity between a pair of SGs. Therefore, we use the following formula to normalize the similarity of a pair of SGs. For the example shown in Fig. 5, the similarity between SGX and SGY is 0.6428.

$$Similarity_{SGX,SGY} = \frac{2 \times |LCS[m, n]|}{m + n}$$

Table 2 shows the pseudo code of the algorithm of calculating the similarity between all comprising SGs in a Web page. Given a Web page (e.g., Fig. 4(A)), all SGs are identified and organized into a structure tree (as shown in Fig. 4(B)). In order to be recursively traversed, such a structure tree is transformed into a right-sibling-left-child binary tree (short for RSLC tree), where the left-most child of a node represents the child node in the structure tree and other child nodes represent the nearest sibling nodes in the structure tree. The generated RSLC tree then becomes the input of the algorithm. As shown in Table 2, the algorithm uses the post-order traversal to recursively navigate the RSLC tree and perform the comparison formulas on each PASG. The output is a list of similarity values between PASGs. We run the algorithm over the structure tree shown in Fig. 4(B). The tracks of the recursive operations are represented by the gray-dotted lines labeled with numbers indicating the traversal orders, as shown in Fig. 4(B).

Table 2
LCS-based similarity algorithm.

INPUT: Right-sibling-left-child binary tree
INITIAL: parent node ← root
CalculateSimilarity (parent node)
1. if (left child node of the parent node exists)
2. then SimilarityCalculate (the left child node)
3. if (the right child node of the parent node exists)
4. then SGX ← parent node
5. SGY ← right child of parent node
6. for (i ← 1 to SGX)
7. do for (j ← 1 to SGY)
8. do if (i = 0 or j = 0) then $ LCS[i, j] = 0$
9. if (i, j > 0 and $x_i = y_j$) then $ LCS[i, j] = LCS[i - 1, j - 1] + 1$
10. if (i, j > 0 and $x_i \neq y_j$) then $ LCS[i, j] = \max(LCS[i, j - 1] , LCS[i - 1, j])$
11. end for
12. end for
13. $Similarity = \frac{2 \times LCS[SGX , SGY]}{ SGX + SGY }$
14. OUTPUT Similarity
15. Similarity ← 0
16. CalculateSimilarity (the right child node)

After calculating similarities between every PASG, the next challenge is to determine the relevance level between each PASG. In general, two SGs with a higher similarity value are more relevant than those with a lower value. However, how could we decide the boundary of being relevant or not automatically?

3.3. Similarity-to-RoF transformation

In this research, we apply the entropy theory to construct a fuzzy membership function that transfers the obtained similarities into *Relevance of Functionality* (RoF). We can then utilize RoF to simulate human sense to quantitatively decide the relevance of two semantic groups using the fuzzy method (Ross, 2004), which has been widely adopted to deal with real-world problems with vague statements.

Recall that we have normalized the similarities between semantic groups. As shown in Fig. 6, we define RoF to divide all possible similarities, from 0 to 1, into four zones: *absolutely irrelevant* (AIZ), *possibly irrelevant* (PIZ), *possibly relevant* (PRZ), and *absolutely relevant* (ARZ). The four zones are delimited by three thresholds: T_{IPA} , T_{IR} , and T_{RPA} .

We realize automatic classifications by constructing a fuzzy membership function using inductive reasoning from a set of training data. First, we define the rules of judging the relevance level of a pair of semantic group based on their similarities:

1. A pair of SGs are highly possible to be considered as relevant by a human being, if their similarity falls into ARZ, i.e., greater than T_{RPA} .
2. A pair of SGs are likely to be considered as relevant by a human being, if their similarity falls into PRZ, i.e., between T_{IR} and T_{RPA} . The certainty level in this zone is proportional to the similarity.

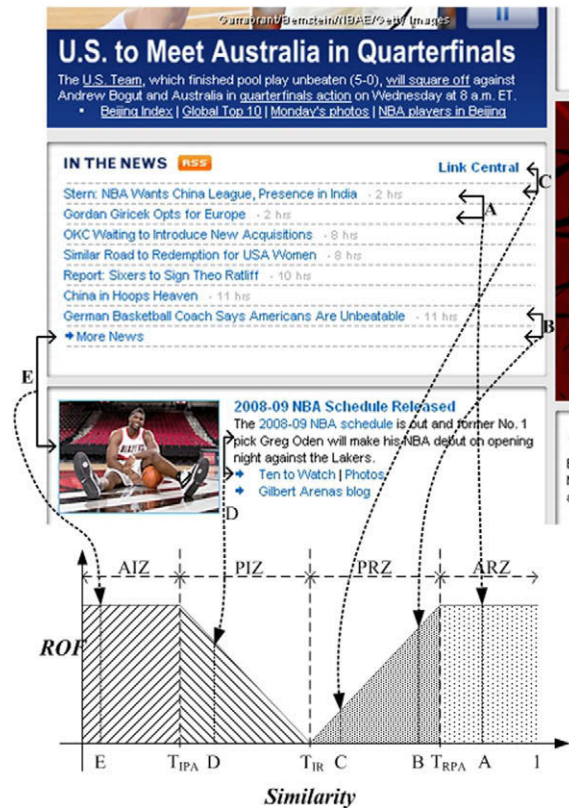


Fig. 6. An example of a fuzzy membership function.

3. A pair of SGs are likely to be considered as irrelevant by a human being, if their similarity falls into AIZ, i.e., less than T_{IPA} .
4. A pair of SGs are highly possible to be considered as irrelevant by a human being, if their similarity falls into PIZ, i.e., between T_{IPA} and T_{IR} . The certainty level in this zone is inversely proportional to the similarity.

Let us take a look at the examples shown in Fig. 6. Identifiers A, B, C, D, and E represent five pairs of PASG in a Web page. Their corresponding similarities are marked by the points on the similarity axis labeled with the same identifiers. Each of the points fall into one of the four zones. According to the rules above, it is easy to decide that PASG A is relevant because its similarity belongs to ARZ, and PASG E is irrelevant because its similarity belongs to AIZ. PASGs B and C belong to PRZ, so that the machine tends to consider that they are relevant but not certain. The uncertainty of PASG C is stronger than that of PASG B. Similarly, PASG D is considered closer to be irrelevant.

3.4. Membership function

Our next step is to construct a fuzzy membership function that automatically transfers the similarities into RoF. We first create an event space by gathering a number of samples (i.e., PASGs). Then we determine the three thresholds (T_{IR} , T_{IPA} and T_{RPA}), which are the boundaries to divide the similarities into the four zones. Among them, T_{IR} is determined first to divide the event space into two zones, relevant zone and irrelevant zone. T_{IPA} and T_{RPA} are then determined to further divide the two zones into four zones. The key to determining each threshold is to calculate the smallest entropy for all possible partitions that can divide the event space into two zones.

For an event space E , an entropy represents the degree of incorrect classification given a partition x as a threshold to divide E into two zones. According to the goal of the threshold, a zone can be partitioned into three ways, where z represents the resulted zone after the partition:

1. If x is set to find T_{IR} , then $z=\{r: \text{relevant zone(RZ)}, i: \text{irrelevant zone (IZ)}\}$
2. If x is set to find T_{IPA} , then $z=\{r: \text{possibly irrelevant zone (AIZ)}, i: \text{absolutely irrelevant zone (PIZ)}\}$
3. If x is set to find T_{RPA} , then $z=\{r: \text{absolutely relevant zone (ARZ)}, i: \text{possibly relevant zone (PRZ)}\}$

Assume that $E = s$. Every e_i carries two parameters (s_i, k_i), where s_i is the similarity and k_i is a boolean value that is true (t for short) if e_i is considered belonging to a partitioned z and false (f for short) if otherwise. Given a partition x in E , N represents the total number of events in E , $n^z(x)$ represents the total number of events in zone z .

$$n_k^z(x) = \begin{cases} n_t^z(x), & \text{the number of events belonging to zone } z \text{ (true events),} \\ n_f^z(x) & \text{the number of events not belonging to zone } z \text{ (false events)} \end{cases}$$

Using the entropy theory, the inductive reasoning formulas can be listed as follows:

$$P_k^z(x) = \frac{n_k^z(x) + 1}{n^z(x) + 1}$$

$$P^z(x) = \frac{n^z(x)}{N}, \quad \text{where } P^r + P^i = 1$$

$$H_{P_t}(x) = -[P_t^r(x) \ln P_t^r(x) + P_t^i(x) \ln P_t^i(x)]$$

$$H_{P_f}(x) = -[P_f^r(x) \ln P_f^r(x) + P_f^i(x) \ln P_f^i(x)]$$

$$H(x) = P^r(x)H_{P_t}(x) + P^i(x)H_{P_f}(x)$$

The inductive reasoning suggests that the best threshold refers to the smallest entropy over all possible partitions. Take T_{IR} as an example. We first set partition x in a way that the first element itself forms a zone and the remaining elements form the other zone. The first entropy $H(x)$ is calculated using the formulas above. Second, the second element is moved from the second zone into the first, and the second entropy is calculated again. Using the same method, the elements in the second zone are gradually moved into the first zone, one at a time, to represent another partition. For each partition, an entropy is calculated. The partition with the smallest entropy decides T_{IR} and divides E into two new event spaces. Similarly, T_{IPA} and T_{RPA} can be deduced using the same procedure from the two obtained new event spaces.

Let us take an example to illustrate the training procedure. Given E shown in Fig. 7(A), 17 balls consist of 8 black balls standing for irrelevant events (false events) and 9 white balls standing for relevant events (true events). To find the threshold T_{IR} , we first calculate an entropy when x is set between ball 1 and 2, (1, 2). Ball 1 forms a zone and balls 2 to 17 form the other zone. The second entropy can be acquired when x is set between (2, 3).

The following list shows the details of calculating the entropy when setting x at (7, 8):

$$\begin{aligned} n_t^{IZ}([7, 8]) &= 6, & n_f^{IZ}([7, 8]) &= 1, & n_t^{RZ}([7, 8]) &= 8, & n_f^{RZ}([7, 8]) &= 2, \\ n^{IZ}([7, 8]) &= 7, & n^{RZ}([7, 8]) &= 10, & P_t^{IZ}([7, 8]) &= \frac{6+1}{7+1} = 0.875, \\ P_t^{RZ}([7, 8]) &= \frac{8+1}{10+1} = 0.818, & P_f^{RZ}([7, 8]) &= \frac{7}{17} = 0.412, \\ P_f^{IZ}([7, 8]) &= \frac{1+1}{7+1} = 0.25, & P_f^{RZ}([7, 8]) &= \frac{2+1}{10+1} = 0.272, \\ P^{RZ}([7, 8]) &= \frac{10}{17} = 0.588, & H_{P_t}([7, 8]) &= 0.436, \\ H_{P_f}([7, 8]) &= 0.519, & H([7, 8]) &= 0.496 \end{aligned}$$

Table 3 shows the partial process of finding T_{IR} . Some entropies of partitions (when x is set at (1, 2), (2, 3), (12, 13), (13, 14), (14, 15), (15, 16) and (16, 17)) have higher values so they are omitted from the table due to the space limitation. As a result, T_{IR} is set between (7, 8) to divide the 17 balls into two zones, RZ and IZ, as shown in Fig. 7(B). Then, we use the same procedure to find T_{IPA} , which classifies the new E , ball 1 to ball 7, into the two zones, AIZ and PIZ. Afterwards, we find T_{RPA} , which classifies the other new E , ball 8 to ball 17, into two zones, ARZ and PRZ. The results are shown in Tables 4 and 5, respectively. After the three thresholds are obtained, we can construct the membership function as shown in Fig. 7(C).

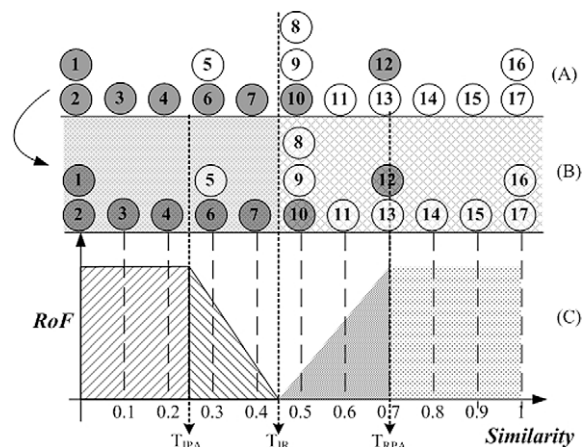


Fig. 7. An example of building a fuzzy membership function from a training data set.

Table 3
The training results of finding T_{IR} .

$x = T_{IR}$	(3,4)	(4,5)	(5,6)	(6,7)	(7,8)	(8,9)	(9,10)	(10,11)	(11,12)
$p_f^{IZ}(T_{IR})$	1.000	1.000	0.833	0.857	0.875	0.778	0.700	0.727	0.667
$p_e^{IZ}(T_{IR})$	0.250	0.200	0.333	0.286	0.250	0.333	0.400	0.364	0.417
$p_f^{RZ}(T_{IR})$	0.400	0.357	0.385	0.333	0.273	0.300	0.333	0.250	0.286
$p_e^{RZ}(T_{IR})$	0.667	0.714	0.692	0.750	0.818	0.800	0.778	0.875	0.857
$p_f^{IZ}(T_{IR})$	0.176	0.235	0.294	0.353	0.412	0.471	0.529	0.588	0.647
$p_e^{RZ}(T_{IR})$	0.824	0.765	0.706	0.647	0.588	0.529	0.471	0.412	0.353
$H_{P_i}(T_{IR})$	0.347	0.322	0.518	0.490	0.463	0.562	0.616	0.599	0.635
$H_{P_j}(T_{IR})$	0.637	0.608	0.622	0.582	0.519	0.540	0.562	0.463	0.490
$H(T_{IR})$	0.586	0.541	0.592	0.550	0.496	0.550	0.591	0.543	0.584

Table 4
The training results of finding out T_{IPA} .

$x = T_{IPA}$	(2,3)	(3,4)	(4,5)	(5,6)
$p_f^{AIZ}(T_{IPA})$	1.000	1.000	1.000	0.833
$p_e^{AIZ}(T_{IPA})$	0.333	0.250	0.200	0.333
$p_f^{PIZ}(T_{IPA})$	0.833	0.400	0.500	0.333
$p_e^{PIZ}(T_{IPA})$	0.333	0.800	0.750	1.000
$p_f^{RIZ}(T_{IPA})$	0.286	0.429	0.571	0.714
$p_e^{RIZ}(T_{IPA})$	0.714	0.571	0.429	0.286
$H_{P_i}(T_{IPA})$	0.366	0.347	0.322	0.518
$H_{P_j}(T_{IPA})$	0.518	0.545	0.562	0.366
$H(T_{IPA})$	0.475	0.460	0.425	0.475

Table 5
The training results of finding out T_{RPA} .

$x = T_{RPA}$	(11,12)	(12,13)	(13,14)	(14,15)
$p_f^{ARZ}(T_{RPA})$	0.400	0.500	0.429	0.375
$p_e^{ARZ}(T_{RPA})$	0.800	0.667	0.714	0.750
$p_f^{PRZ}(T_{RPA})$	0.286	0.167	0.200	0.250
$p_e^{PRZ}(T_{RPA})$	0.857	1.000	1.000	1.000
$p_f^{ARZ}(T_{RPA})$	0.400	0.500	0.600	0.700
$p_e^{PRZ}(T_{RPA})$	0.600	0.500	0.400	0.300
$H_{P_i}(T_{RPA})$	0.545	0.617	0.603	0.584
$H_{P_j}(T_{RPA})$	0.490	0.299	0.322	0.347
$H(T_{RPA})$	0.512	0.458	0.491	0.512

In the real case, we will get the three thresholds as $T_{IPA} = 0.203$, $T_{IR} = 0.521$ and $T_{RPA} = 0.765$. We will explain more in the experiment section.

3.5. Content adaptation

As aforementioned, component resizing and layout rearrangement are the two main methods used in content adaptations. The proportion of the screen size of a PC against the mobile device is usually used for the component resizing. However, an unreasonably high proportion may lead to unreadable resolutions of presentation objects. He et al. (2007) propose a method of estimating resizing proportions using a fuzzy model. In this project, we aim at exploring a proper layout rearranging strategy to adapt a Web page. To simplify the operations, from repeatedly horizontal-and-vertical bilateral scrolling to up-down scrolling, the width of a Web page should be reduced by column-wise adjustments (CWA for short), which is the means to change the arrangements of presentation objects from the original horizontal layout to a vertical layout. However, performing CWAs should take semantic coherence into account in order to avoid semantic distortions.

We designed an adaptation algorithm based on our entropy-based fuzzy reasoning method. The four proposed zones stand

for the tolerance of distortion, i.e., the risk degree of distortion, as performing CWAs at a PASG. According to our previous study, CWAs should be performed when one of the following two conditions hold:

- C1: Users ask for adaptations;
- C2: The width of a Web page is greater than that of the receiving device.

Suppose that the input is a given Web page whose all possible PASGs and their similarities have been computed. We start to perform CWAs at all PASGs belonging to AIZ, i.e., selecting all the PASGs with similarities falling in the interval $([0, 0.23])$. We consider AIZ first because the PASGs in AIZ are irrelevant. After this step, we check the two conditions. If one of them holds, the adaptation should be continued in PIZ. Otherwise, the algorithm stops and the Web page is suitable for display.

Since the confidence level of the PASGs in PIZ being considered as irrelevant is weaker than that in AIZ (that is, RoF starts to decrease), the algorithm selects PASGs by viewing an interval in PIZ as a unit to gradually perform CWAs. Both the conditions should be checked after all the PASGs in the interval have been adapted. If both C1 and C2 do not hold, the algorithm should be stopped and deliver the adapted Web page to users. Otherwise, the interval x should move to the right to select the next set of PASGs in PIZ to perform the CWAs. The procedure should continue until the interval has been moved to $[0.521 - x, 0.521]$, i.e., the adaptation should continue in the next zone, PRZ. Note that the interval x is a constant that is experimentally set as 0.0159 each time the PASGs are selected in PIZ. This is because all the PASGs in PIZ still are considered irrelevant.

Similar to the adaptation in PIZ, the algorithm selects PASGs to perform CWAs by viewing an interval x in PRZ. However, since the PASGs in PRZ start to be considered as possibly relevant, the interval x should be dynamically changed according to the number of CWAs performed. The interval should be gradually reduced as it is moved to the right. We designed a logarithm function to dynamically change the interval as follows:

$$x = \eta \log_b \left(\frac{j}{j-1} \right)^a$$

where j is an increasing value and η , a , b are constants that are experimentally set as $\eta = 0.15$, $a = 5$ and $b = 1000$. The interval x will be gradually reduced following the increasing value j as shown in Fig. 8. In other words, the number of selected PASGs will be decreased according to the increasing similarities. In addition to the dynamic interval, it is also different from adaptations in PIZ when the adaptation need merely depends on C1. This is because the PASGs in PRZ are generally relevant. The risk of distortions should be estimated by users. Therefore, the adaptation starts from setting interval x as $[0.521, 0.521] + (x)$ and initial value j as 3, so that the

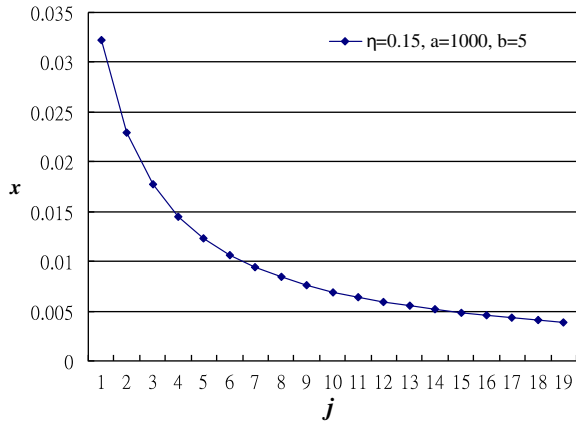


Fig. 8. Change of interval in PRZ.

PASGs with similarities fall in the interval $[(0.521, 0.521 + 0.15 \log_{1000}(1.5)^5)]$ should be adapted by performing CWAs. Afterwards, we check $C1$ to determine whether the Web page should be further adapted. If not, the adaptation is finished. Otherwise, the adaptation should move on to the next right interval and increase j . The procedure goes on until $C1$ does not hold or the interval has been moved to $([0.765 - x, 0.765])$. The former means that the adaptation is sufficient in PRZ; the latter means that all the PASGs in PRZ have been adapted. The Web page does not allow any adaptation since the rest of PASGs belongs to ARZ, that is, they are generally relevant. Table 6 shows the pseudo code of our proposed algorithm. For illustration purpose, we use $([x1, x2])$ to stand for the interval x .

4. Content adaptation engine design

We constructed a functionality sense-based content adaptation (FSCA) engine, which overall architecture is shown in Fig. 9. Instead of deploying our FSCA engine in either server side or client side, we decided to implement it as a Web service, so that the content adaptation can be realized transparent to any platform and devices.

Table 6
Content adaptation algorithm.

1.	INPUT: A quantified Web page
2.	If ($C1=true$ or $C2=true$)
3.	then Do the CWAs at PASGs with similarity s_i in $([0, 0.203])$
4.	$x1 \leftarrow 0.203$
5.	$x2 \leftarrow x1 + 0.0159$
6.	while ($(C1=true$ or $C2=true)$ and $x2 \leq 0.521$)
7.	Do Do the CWAs at PASGs with similarity s_i in $([x1, x2])$
8.	$x1 \leftarrow x2$
9.	$x2 \leftarrow x1 + 0.0159$
10.	End while
11.	$j \leftarrow 3$
12.	$x1 \leftarrow 0.521$
13.	$x2 \leftarrow 0.15 \log_{1000}(1.5)^5$
14.	while ($C1=true$ and $x2 \leq 0.765$)
15.	Do Do the CWAs at PASGs with similarity s_i in $([x1, x2])$
16.	$j++$
17.	$x1 \leftarrow x2$
18.	$x2 \leftarrow x1 + 0.15 \log_{1000} \left(\frac{j}{j-1} \right)^5$
19.	End while
20.	OUTPUT: The adapted Web page

As shown in Fig. 9, the FSCA engine comprises three major modules: (1) *similarity quantization*, (2) *similarityToRoF translator*, and (3) *adaptation machine*. The FSCA engine acquires an HTML document from Web servers as input, and translates it into a structure tree. The similarity quantization module applies the LCS algorithm to calculate the similarity between adjacent object groups, by comparing their attributes and structures. The similarities are the inputs of the *similarityToRoF* translator and are translated into corresponding zones by using the fuzzy membership function to acquire related RoFs. The *similarityToRoF* translator simulates human sense to decide the relevance of two objects by the fuzzy method. Finally, based on the obtained RoFs information, the adaptation machine selects a proper strategy for rearranging layout of contents to avoid the distortion as far as possible.

As we implemented the FSCA engine as a Web service, Web servers send HTML pages to FSCA service together with the receiving destination information. Our service detects the receiving context using the techniques we present in Yang and Shao (2006), Yang et al. (2008). The extracted data guides the FSCA service to adapt the content accordingly. Finally, the FSCA service forwards the adapted content to the receiving device.

In this project, we classify commonly used devices into four types: Desktop, Potable PC, PDA and smart phone. In general cases, desktops and notebooks do not need adaptations for usual Web pages. However, ASUS EeePC is a portable PC whose screen size is only 7 in. It may need some light layout adaptation for the best presentation. By observations, we found that it is sufficient that the adaptation only selects PASGs for AIZ zone. For the PDA type, typical product models are HP iPAQ 5400 pocket PC with a screen size of 240×320 pixels and Blackberry 8707g with a screen size of 240×320 pixels (2.4 inch). Obviously PDAs require more intense adaptation, i.e., PIZ zone adaptations. For the smart phone type, Mio 8390 with a screen size of 176×220 pixels (2.2 inches) is a practical product model. Its screen size is the smallest so it requires the most complicated adaptation.

5. Experiments

We designed and conducted three series of experiments to test our membership function, content adaptation, and user acceptance.

5.1. Membership function-related experiment

Since the membership function is obtained from a training data set using inductive reasoning, it is important to know how many samples are sufficient to stand for the real-world situation. The three thresholds were recorded, each time we added 50 samples into E until the size reached 1000. In Fig. 10, the x -axis refers to the size of E and the y -axis refers to the values of the three thresholds. It can be observed that a stable state is reached after the size of E is around 800, when the three thresholds read as: $T_{IPA} = 0.203$, $T_{IR} = 0.521$ and $T_{RPA} = 0.765$.

To verify the performance of the membership function, we picked up 15 Web pages, excluding from the training data set, as the test data set including 432 PASGs associated with the similarities computed by the LCS algorithm. The relevance of the collected PASGs is manually determined by the expert's judgments and is taken down by the proportions as follows:

$$P_{relevant}(x) = \frac{Amount_{relevant}(x)}{Amount_{relevant}(x) + Amount_{irrelevant}(x)}$$

$$P_{irrelevant}(x) = \frac{Amount_{irrelevant}(x)}{Amount_{relevant}(x) + Amount_{irrelevant}(x)}$$

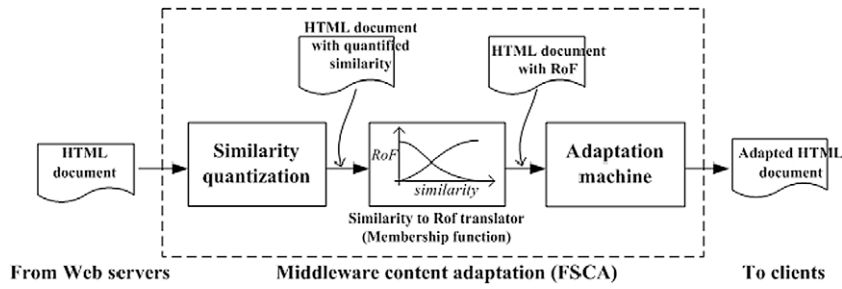


Fig. 9. System architecture of FSCA.

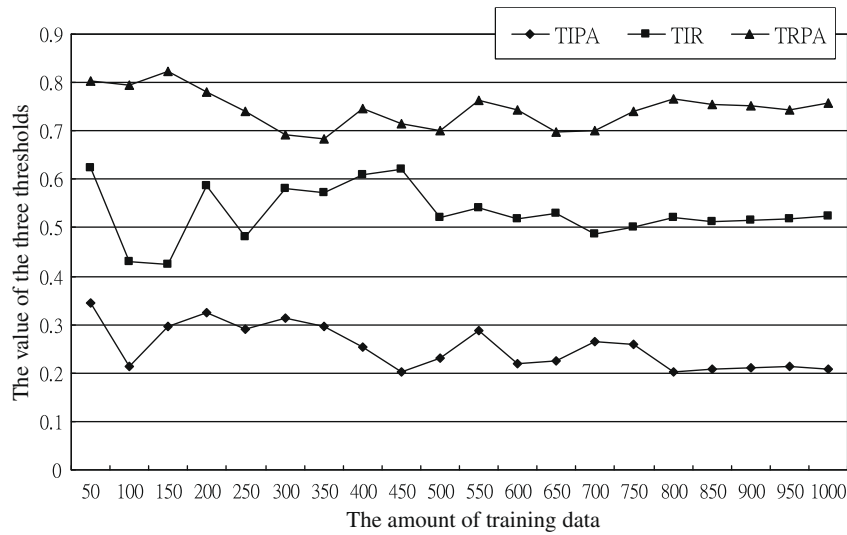


Fig. 10. Training records for three thresholds.

In the two formulas, $P_{relevant}(x)$ represents the proportion of relevance among the similarity interval x . $Amount_{relevant}(x)$ represents the number of the relevant events among x ; $Amount_{irrelevant}(x)$ represents the number of the irrelevant events among x . In Fig. 11, the x-axis refers to similarity, where 0.05 is the interval unit of calcu-

lating the proportions, and the y-axis refers to the propositions of relevance and irrelevance. As shown in Fig. 11, the results show that the distribution is very close to the membership function built. The correct rate of the similarities belonging to ARZ is 97.1%; the correct rate of the similarity belonging to AIZ is 99%.

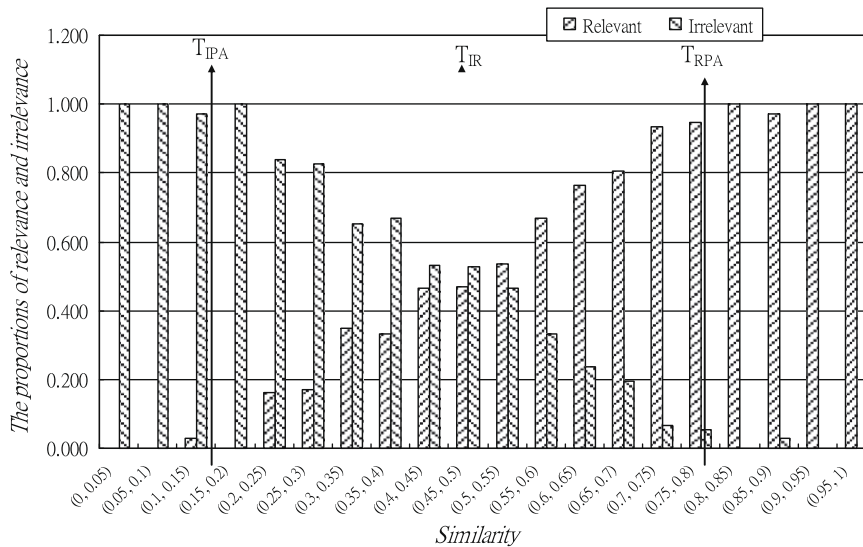


Fig. 11. The results of testing data.

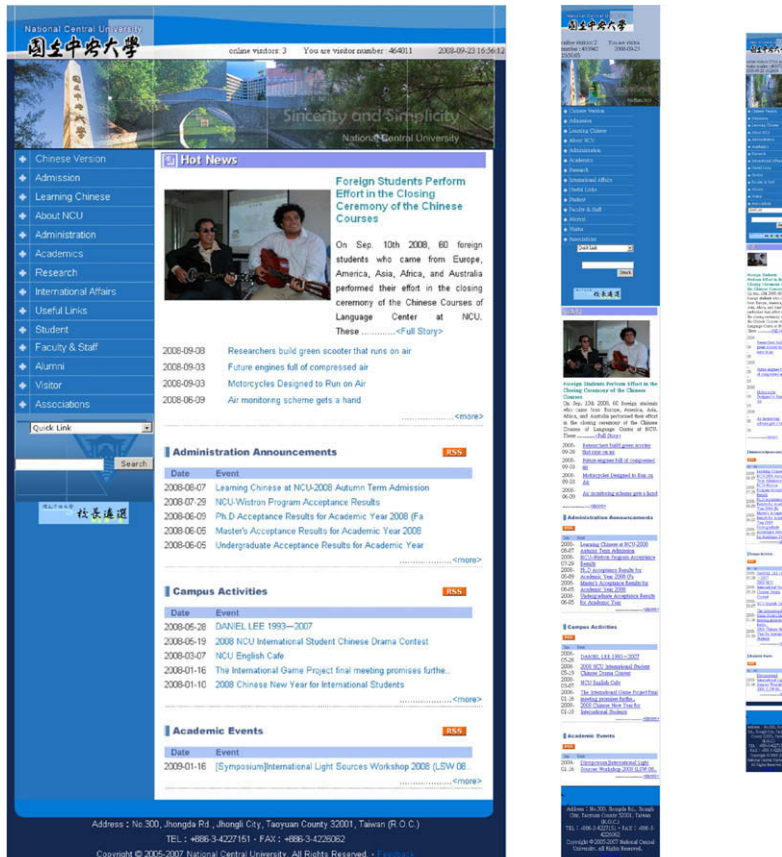
5.2. Adaptation experiment

We selected a number of running Web pages to perform the adaptation for three types of devices: portable PC, PDA, and smart phone. We separately recorded the number of the adjustments

(A.NO. for short) and the number of the distortions (D.NO. for short), and compare them with the results of the single column-wised adaptation. The results are listed in Table 7, where the 18 randomly selected Web pages are classified into 5 classes including University (U), Business (B), Sport (S), News (N) and portal (P).

Table 7
The verification of FSCA.

Class	Website name	RoF based content adaptation						Signal column based adaptation	
		Portable PC		PDA		Smart phone		A. NO.	D. NO.
		A. NO.	D. NO.	A. NO.	D. NO.	A. NO.	D. NO.		
U	NCU	4	0	8	0	8	0	75	17
	NCTU	3	0	5	0	5	0	63	13
	Harvard	5	0	6	0	7	0	55	14
B	Berkeley	2	0	3	0	3	0	39	2
	ASUS	2	0	2	0	2	0	92	19
	Acer	2	0	2	0	2	0	32	5
	Amazon	2	0	3	0	3	0	73	8
S	Ebay	2	0	2	0	3	0	53	1
	Celtics	2	0	3	0	6	1	92	25
	US open	2	0	3	0	4	0	46	12
N	Wimbledon	3	0	5	1	5	0	72	14
	Local News UDN	4	0	6	1	9	1	137	19
	Google News	1	0	6	1	6	0	44	3
P	CNN	3	0	6	1	14	3	153	29
	Kaohsiung city government	2	0	2	0	4	1	94	16
	Travel (CTIN)	2	0	5	0	10	2	284	26
	Facebook	2	0	2	0	2	0	28	5
	NCU Blackboard	3	0	3	0	3	0	11	3
Total		46	0	72	4	96	8	1443	231
D. NO.		0%		5.56%		8.3%		16%	
A. NO.									



(a) NCU Web page for desktop PCs (b) For PDAs (c) For smart phones

Fig. 12. (a) NCU Web page for desktop PCs, (b) for PDAs, (c) for smart phones.

For the type of Portable PC, AIZ class adaptations are sufficient, so that the proportion of the distortions is around 0%. For the type of PDA, PIZ class adaptations are used, so that the proportions of the

distortions increase to 5.56%. For the type of smart phone, the adaptations go into PRZ. The proportions of the distortions thus increase to 8.3%. Comparing to the signal column-wise adaptation



(a) The BBC Web page for desktop PC (b) For portable PC (c) For PDA

Fig. 13. (a) The BBC Web page for desktop PC, (b) for portable PC, (c) for PDA.



Fig. 14. The presentation of the NCU Web page without CA.



Fig. 15. The presentation of the Harvard Web page through FSCA.

In regard to following perspectives, please give the satisfaction scores of browsing the original pages and adapted pages respectively:

P1. Stylus movements needed for browsing the pages.
 P2. Easiness to find the specific information in the pages.
 P3. Presentation of the content in the pages.
 P4. Overall browsing experience for the whole pages.

		Please fill in the form		
		Dissatisfied (1~3)	Average (4~6)	Satisfied (7~9)
For Original Pages	P1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	P2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	P3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	P4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
For Adapted Pages	P1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	P2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	P3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	P4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Extra Comments				

Fig. 16. User study questionnaire.

that all of adjacent presentation objects are adapted without concerning semantic coherence, both the numbers of the adjustment and the distortions decrease. The proportions of the distortions decrease to 16% as well. The experiment proves that our method avoids a significant amount of useless adjustments.

5.3. Adaptation result

We used three real cases to perform FSCA and the results are presented as follows. For the first case, we presented the Web page

of the National Central University on desktops, PDAs and smart phones. In Fig. 12, (a) is the original Web page suitable for desktop display; (b) and (c) are two individual adapted versions for PDAs and smart phones. Beside the width, they are quite similar, since most of the similarities distribute at the two ends of the membership function.

For the second case, we presented the BBC Web page with the three versions. Fig. 13(a) shows the original version presented on a desktop. Fig. 13(b) presents the adapted version for a portable PC. Comparing (b) with (a), it is easy to see that the satisfactory version for portable PCs merely needs a few adjustments. On the other hand, Fig. 13(c) is the adapted version for PDAs with a number of adjustments because of their much smaller screen sizes.

Finally, we specifically presented the NCU Web page on the smart phone without CA, as shown in Fig. 14, and with CA, as shown in Fig. 15. In Fig. 14, the original Web page requires horizontal and vertical scrolling highlighted by circles, and is not capable of integral presentations. The adapted Web page can be presented better, as shown in Fig. 15, and merely requires vertical scrolling. Obviously, our FSCA not only simplifies operations but also maintains the semantic coherence.

5.4. Questionnaire design and survey

We designed a questionnaire to evaluate the testers' satisfaction levels, as shown in Fig. 16. In this questionnaire, we focused on investigating four perspectives related to testers' browsing experience. P1 reflects the frequency of testers' actions relative to their browsing habits, such as single-dimensional scrolling or bi-dimensional scrolling. P2 intends to find out whether the testers can easily find the specific information on Web pages. P3 is concerned with the content presentation such as the layout or some

Table 8 Scores of user satisfaction.

Score	Question		Question		Question		Question	
	1	2	3	4	5	6	7	8
	Original pages	Adapted pages	Original pages	Adapted pages	Original pages	Adapted pages	Original pages	Adapted pages
1			4				4	
2	6		5			1	3	
3	11		12	2	6	2	11	2
4	5		9	1	4	1	7	
5	8	3		5	8	4	5	5
6		5		8	7	7		7
7		10		8	2	10		6
8		11		4	3	3		7
9		1		2		2		3
Average	3.5	7.1	2.9	6.3	5.1	6.2	3.2	6.6

dynamic functions of tested Web pages. If the layout is not neat or the dynamic functions are ruined, the user may feel confused thus does not feel satisfied. P4 is an overall investigation of testers' browsing experience using our system. Finally, the testers may give some extra comments about using our system. In regard to these perspectives, the testers were asked to give scores of satisfaction levels after browsing the selected Web pages. Valid scores are between [1, 9], with 1 being the lowest score and 9 being the highest score.

After all testers finished the questionnaire, we summed up the total scores and calculated the average scores. The results are shown in Table 8. Each cell in the table represents the number of testers who gave the corresponding scores for the given perspective. The testing scores show that all testers prefer our adapted pages to the original pages.

We also received some comments, which revealed two situations about content presentation. One is that some images become very blurred and hard to be recognized. The other is that some contents overlap with each other. In a Web page, a word may be written over another word. We found that it is caused by broken <DIV> tags. Since these situations are rare, the overall user satisfaction is highly positive.

6. Conclusions

In this paper, we present a novel concept to implement content adaptation with a lower distortion rate. Our method better decides the relevance between presentation objects by using the fuzzy reasoning technique to simulate human sense. In addition, by implementing the content adaptation engine as a service, users can acquire the adaptation services through the Internet without changing any client side and server-side settings. As new portable devices, such as iPhone and google phone, are continually launched over the market, it is critical to continue to study more effective and efficient content adaptation techniques.

For our future study, we plan to construct an open-source service site to allow users and machines to exploit our content adaptation services. The expected user experiences will also allow us to further validate and enhance our approach.

Acknowledgements

This work is supported by National Science Council, Taiwan under grants NSC95-2520-S-008-006-MY3 and NSC 96-2628-S-008-008-MY3.

References

- Berhe, G., Brunie, L., Pierson, J.M. (2004). Modeling service-based multimedia content adaptation in pervasive computing. In *Proceedings of the first conference on computing frontiers* (pp. 60–69).
- Burzagli, L., Emiliani, P. L., & Gabbanini, F. (2009). Design for all in action: An example of analysis and implementation. *Expert System with Applications*, 36(2), 985–994.
- Chen, L. Q., Xie, X., Ma, W. Y., Zhang, H. J., Zhou, H. Q., Feng, H. Q. (2002). DRESS: A slicing tree based web representation for various display sizes. Technical report MSR-TR-2002-126, Microsoft Research.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2001). *Introduction to Algorithms* (2nd ed.). Cambridge, MA, USA: The MIT Press.
- Google.com. Google. Available from: <http://www.google.com/>.
- He, J., Gao, T., Hao, W., Yen, I. L., & Bastani, F. (2007). A flexible content adaptation system using a rule-based approach. *IEEE Transactions on Knowledge and Data Engineering*, 19(1), 127–140.
- Hua, Z., Xie, X., Liu, H., Lu, H., & Ma, W. Y. (2006). Design and performance studies of an adaptive scheme for serving dynamic web content in a mobile computing environment. *IEEE Transactions on Mobile Computing*, 5(12), 1650–1662.
- IBM. WebSphere Transcoding Publisher. Available from: http://www-306.ibm.com/software/pervasive/transcoding_publisher/.
- Lee, Y. W., Chandranmenon, G., Miller, S. C. (2003). GAMMA: A content adaptation server for wireless multimedia applications. Bell-Labs, Technical Report.
- Lemlouma, T., Layaida, N. (2004). Context-aware adaptation for mobile devices. In *Proceedings of 2004 IEEE international conference on mobile data management* (pp. 106–111).
- Oracle. Oracle application server wireless. Available from: <http://www.oracle.com/technology/tech/wireless/index.html>.
- Phan, T., Zorpas, G., Bagrodia, R. (2002). An extensible and scalable content adaptation pipeline architecture to support heterogeneous clients. In *Proceedings of the 22nd international conference on distributed computing systems* (pp. 507–516).
- Ramaswamy, L., Iyengar, A., Liu, L., & Douglass, F. (2005). Automatic fragment detection in dynamic web pages and its impact on caching. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), 859–874.
- Ross, T. J. (2004). *Fuzzy Logic with Engineering Applications* (2nd ed.). England: John Wiley and Sons Ltd.
- Sun. Java system portal server mobile access. Available from: http://www.sun.com/software/products/portal_srvr/index.xml.
- Tidy. Tidy. Available from: <http://www.tidy.org>.
- Tsai, Y. C., Cheng, C. H., & Chang, J. R. (2006). Entropy-based fuzzy tough classification approach for extracting classification rules. *Expert Systems with Applications*, 31(2), 436–443.
- W3C. W3C candidate recommendation: Cascading style sheets level 2 revision 1 (CSS 2.1) specification. Available from: <http://www.w3.org/TR/CSS21>.
- Yahoo.com. Yahoo. Available from: <http://www.yahoo.com>.
- Yang, S. J. H., & Shao, N. W. Y. (2006). An ontology based content model for intelligent web content access services. *International Journal of Web Service Research (JWSR)*, 3(2), 59–78.
- Yang, S. J. H., Zhang, J., & Chen, I. Y. L. (2007). Ubiquitous provision of context-aware web services. *International Journal of Web Services Research (JWSR)*, 4(4), 83–103.
- Yang, S. J. H., Zhang, J., & Chen, I. Y. L. (2008). A JESS enabled context elicitation system for providing context-aware web services. *Expert Systems with Applications*, 34, 2254–2266.
- Yang, S. J. H., Zhang, J., Chen, R. C. S., & Shao, N. W. Y. (2007). A UOI-based content adaptation method for improving web content accessibility in the mobile internet. *ETRI Journal*, 29(6), 794–807.
- Zhang, J., Zhang, L.-J., Quek, F., & Chung, J.-Y. (2005). A service-oriented multimedia componentization model. *International Journal of Web Services Research (JWSR)*, 2(1), 54–76. January–March.