An Adaptive Web Content Delivery System

Jinlin Chen, Yudong Yang, and Hongjiang Zhang

Microsoft Research, China
5F, Beijing Sigma Center, No. 49, Zhichun Road,
Beijing 100080, P.R.China
{i-jlchen, i-ydyang, hjzhang}@microsoft.com

Abstract. The desktop-centric design of most of the current web contents pose many difficulties for pervasive browsing. In this paper, we present our study on the problem to support pervasive browsing in the heterogeneous environment of today' s Internet. A system solution – Adaptive Web Content Delivery (AWCD), is presented to overcome the problems existed in present web infrastructure. The system designed is extensible for the further development of Internet. The two major subsystems of AWCD, client profile learning and adaptation, are described in detail. Experiment results of our system are also shown.

1 Introduction

Today, a variety of appliances are emerging into mainstream Internet appliances, which has brought a large variety in users' preferences in browsing even the same web page. In addition, network connections are also quite different. All these variations require that web pages be prepared suitable for the client. However, most of the current Internet servers still provide the same content to all the clients without considering the variations of client environment, which has caused frustration to many users.

Many people have addressed this problem and presented solutions. SpyGlass [1], OnlineAnywhere [2], QuickWeb [3], FastLane [4], TranSend [5], ProxyNet [6], Digestor [7], and Mobiware [8] are commercial products that adapt web content. However, most of them only make adaptation under special conditions due to the lack of structural information of HTML content, and many of them focus on image conversion. Some projects [9] tried to extract structural information from HTML tag. However, this does not work effective because HTML was designed for content presentation.

In this paper, we present a new approach, Adaptive Web Content Delivery (AWCD), to provide a customized and adaptive Web service for clients. With AWCD, a web user can get content most appropriate to one' s device and preference. And the bandwidth consumption can also be minimized. AWCD provides an overall solution to the existing problems of web content delivery.

In Section 2, we present the design of our prototype of AWCD system. This is followed by discussions on the client profile learning subsystem and adaptation subsystem in Section 3 and Section 4 respectively. In Section 5, we present some experimental results of our system. And conclusions are given in Section 6.

2 System Architecture of AWCD

The basic system architecture of AWCD is shown as Fig. 1.The Client profile learning subsystem discovers the client environment. Adaptation subsystem makes adaptation decisions and generates new web content according to the decision results and various adaptation components.

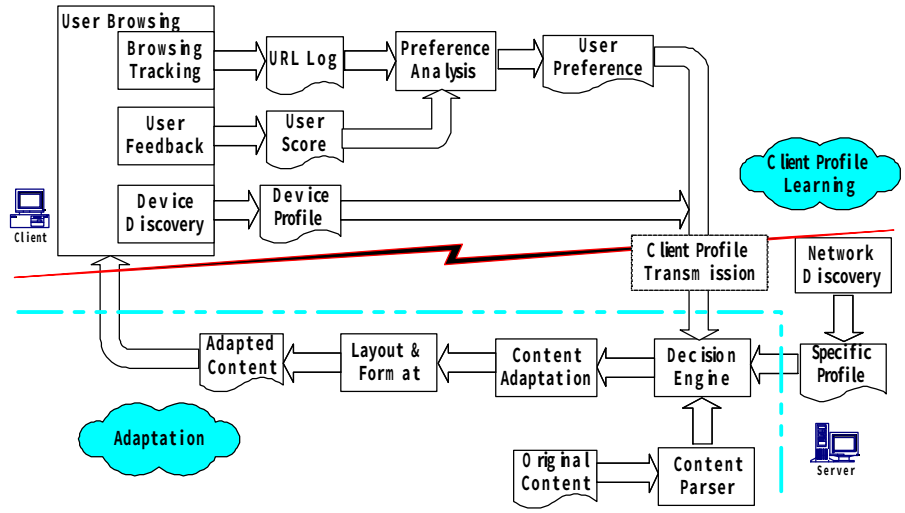

Fig. 1. Basic system architecture of AWCD

Browsing Tracking module tracks users' behavior and obtains user' s browsing paths and times. This is realized by embedding a tracking component in the browser. **User Feedback** module gets user preference by implementing an embedded component of the browser. **Preference Analysis** module analyzes users' preferences according to the browsing tracking results and users' feedback. **Device Discovery** module discovers the hardware and software environments of the client device. **Client Profile Transmission** module transmits user preference and device capability to the server by extending the client-specific web services [11]. **Network Discovery** module discovers network-related parameters, including effective bandwidth, latency, error rate, etc. In our system, we have implemented a TCP level network discovery module. **Content Parser** module parses the requested contents and builds the structural representations of them. **Decision Engine** module collects client profile information from discovery module and contents' data from content parser to determine adaptation instructions and layout rules that will generate the most user-satisfying results in most efficient ways. **Content Adaptation** module consists of filters to perform conversion, summarization and substitution of contents according to the instructions given by the decision engine. **Layout & Format** module reorganizes and generates the final contents to be delivered to end-users.

3 Client Profile Learning

Although many papers [5][10] have discussed the variations of client environment, no one can provide an effective and self-contained way in describing the client environments. Concerning this, we have studied the variations of client environment according to the system framework of Internet, and try to give a logic and systematic description, which focuses on the following dimensions:

Network Characteristics include network bandwidth, latency, error rate, and congestion rate. **Processor Module** includes Hardware capability and software type. **User Interface** includes input/output device and browser information. **User Preference** includes accept waiting time, content preference, media preference, delivery preference, presentation preference, accessibility factor.

We transmit client profiles by extending the client-specific web services, which take client properties as "User Agent Attributes". In our approach we extend "User Agent Attributes" to "Client Attributes" which describe client. Client Attributes are sent through the HTTP User-Agent header. They are added to the normal User-Agent header using a comment field.

4 Adaptation

The kernel of adaptation is Decision Engine. It first retrieves and compares content attributes against client profile to decide what possible operations could be done. It then evaluates the cost of an adaptation process. Quality is evaluated by the quality of content and the user waiting time. Let X^m be media object m, X_i^m be the i^{th} version of X^m, $S(X_i^m)$ be the size of X_i^m, the size of the delivered contents is

$$S_T = \sum_{\text{For selected}i,m} S(X_i^m) \tag{1}$$

Currently we define a simple cost function: a service is acceptable when all contents can be delivered within maximum waiting time specified by users or else the service is not acceptable. Let B_A be the average network bandwidth, D_{RTT} be the network roundtrip time. Assume that transcoding processes and fetching contents to the server could be done in negligible time. The user waiting time for the delivered contents is

$$W_T = 2 \cdot D_{RTT} + \frac{S_T}{B_A} \tag{2}$$

The goal of optimization is then

$$W_T \le T_{MaximumWait} \tag{3}$$

Where T_{maxWait} is the maximum waiting time that a user can accept. We choose a bottom-up approach to do the optimization because the HTML contents could not be well structured.

The optimization starts from a whole collection of detected objects within the document and the first step is to estimate starting W_T. The following step is an iterative

checking and trimming procedure and its stop condition is either (3) is satisfied or there does not have any content we can trim.

5 Experimental Results

We have implemented an experimental system to test our ideas. Our client devices include Windows CE based Palm-size PC, Hand-held PC, Laptop, Desktop. Network connections range from 19.2k wire-line modem to 100M LAN. The experimental data are real web pages from popular Web sites such as MSN. Our demonstration results can be accessed at http://www.research.microsoft.com/mcomputing/acd.
In an average, the transferring time consumed by an adapted page is only about 30 percent of that of the original one. And the layout of the new web page is also more suitable to client device.

6 Conclusion

In this paper we have presented AWCD, an overall adaptation system for delivering Web contents. AWCD can effectively overcome the existing problems in Web content delivery. Compared to other systems, our approach is more robust and extensible.

References

1. Spyglass, "White Paper of Prism 2.2", http://www.spyglass.com/images/Prism22.pdf
2. OnLineAnyWhere, FlashMap, http://www.onlineanywhere.com
3. QuickWeb of Intel Crop. http://www.intel.com/quickweb
4. FastLane of Spectrum Information Technologies http://stage.acunet.net/spectrum/index.html
5. Fox A., Gribble S.D., Chawathe Y., Brewer E.A.: Adapting to Network and Client Variation Using Infrastructural Proxies: Lessons and Perspectives. IEEE Personal Communication (1998), Vol 5(4) 10 – 19
6. ProxiNet, ProxiWare, http://www.proxinet.com/products_n_serv/proxiware/
7. BickMore T.W., Schilit B.N.: Digestor: Device-independent access to the World Wide Web, Proc. of the 6th International World Wide Web Conference. (1997) 655–663
8. Angin, O., Campbell, A.T., Kounavis, M. E., Liao, R.-F.: The Mobiware Toolkit: Programmable support for adaptive mobile networking. IEEE Personal Communication (1998), Vol 5(4) 32–43
9. Lim, S. J., Ng, Y. K.: An Automated Approach for Retrieving Hierarchical Data from HTML Tables. In Proceedings of the Eighth International Conference on Information and Knowledge Management (ACM CIKM'99). Kansas City, MO, USA. (1999) 466-474
10. Smith, J.R., Mohan, R., Li, C.S.: Scalable Multimedia Delivery for Pervasive Computing. ACM Multimedia. (1999) 131–140
11. Client-Specific Web Services by Using User Agent Attributes. W3C Note. http://www.w3.org/TR/NOTE-agent-attributes-971230.html