Adaptive Delivery of HTML Contents

Yudong Yang, Jinlin Chen, and Hongjiang Zhang
Microsoft Research China
{ i-ydyang, i-jlchen, hjzhang }@microsoft.com

Introduction

As Internet becomes the most important information source, people desire to access Internet contents anywhere, anytime with any devices. However, the increasing diversity and heterogeneity of contents, client devices and network conditions combined with individual preferences make "one content fits all needs" impossible. Adaptive content delivery is the system technology that delivers contents dynamically according to the changing situations. Bulky contents are picked to fit users' preferences and condensed to show better on smaller devices with slow connections. Internet accessing becomes faster, more reliable and more economical. Most of our work is focused on two areas: new algorithm to extract structural contents from existing web pages and system technologies to build adaptive content delivery services. Some of the ideas are come from works like[1][2].

System Architecture

1. System Architecture

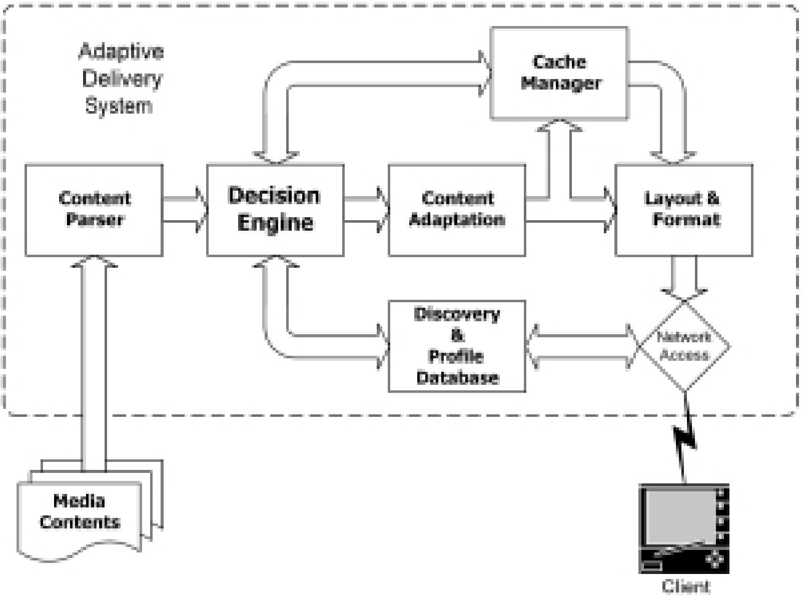

Figure 1: System architecture

Figure 1 shows our system architecture where ***Discovery*** module is a self-improved agent that gathers and maintains information from clients, system and network. ***Content adaptation*** module consists of filters to convert, summarize and substitute contents. ***Layout and format*** module organizes and generates final contents to be delivered. ***Cache manager*** stores frequently used adaptation results for efficiency of process.

2. Layout Based Content Extraction Algorithm

Content parser extracts structural info of contents. HTML lacks the ability of representing semantic related contents because it was designed to take both structural and presentational capability in mind and these two are not clearly separated. Further misuses of structural

HTML tags for layout purpose make the situation even worse. This defeats tag-based content extracting methods[3]. Here we presented a novel approach that is based on layout information of web contents. The algorithm parses HTML pages to extract simple contents and their appearance properties first and then it tries to group these simple contents to *group* objects if they will be rendered in the same paragraph by web browsers. The following step is a pattern discovering procedure to find appearance patterns among those position-related contents. Patterns that cover most contents are considered to be structural information and these covered contents are grouped to *list* objects. These two kinds of objects are the basic of content adaptation. We have tried this algorithm on many popular websites (Yahoo, Altavista, MSN, etc) and it is shown to be very effective because better designed web pages always have similar information organized in similar appearances and close positions.

3. Decision Engine

Decision engine decides what contents and in what appearances and what ways these contents will be sent to end users. It collects information from other modules to figure out adaptation instructions and layout rules that will generate the most user-satisfying and efficient results. It consists of four stages:

1. **Rule based decision** to decide what adaptation methods could or must be done by comparing contents' attributes against user's profile and device's capabilities.
2. **Preference and prediction guidance** to assign priority values to contents according to user's preferences and probabilities from access prediction.
3. **Layout masking** to decide if there is a better layout scheme for displaying these contents on current device by choosing from some layout templates that may best present these contents on the device.
4. **Trade-off optimization** to select final contents and adaptation methods. In this stage we need to balance between quality of contents and the cost to reach it. Here quality is the amount of contents user receives and the encoding quality of contents, and cost includes the complexity to do content adaptation and the expenses user must pay to get these contents.

Experiment

We implemented an experimental system to test our ideas. Our client devices includes Windows CE based Palm-size PC (240X320 color/gray), Hand-held PC (640X200 color), and a software emulator which can emulate arbitrary device/network parameters. These devices are connected through 33.6k modems. Experimental data are downloaded web pages from popular sites like Yahoo, Excite, Altavista, MSN, CNN, etc. Our demonstration and results is available at **http://www.research.microsoft.com/research/china/mcomputing/acd**.

Future Work

Toward high quality ubiquitous Internet access, there are still many untouched issues like online user modeling, QoS evaluation, general hierarchical contents representation and effective delivery methods. We will cover these in future work.

References

1. R. Han, P. Bhagwat, *etc*(1998), Dynamic Adaptation in an Image Transcoding Proxy for Mobile Web Browsing, IEEE Personal Communication
2. W.Y. Ma, I. Bedner, *etc*(2000), A Framework for Adaptive Content Delivery in Heterogeneous Network Environments. MMCN00, San Jose, USA
3. S.J. Lim and Y.K. Ng(1999), WebView: A Tool for Retrieving Internal Structures and Extracting Information from HTML Documents, DASFAA'99, Kansas,USA