Adaptive Service Composition for Meta-searching in a Mobile Environment

Ronnie Cheung, Hassan B. Kazemian

Adaptive Service Composition
for Meta-Searching in a Mobile Environment

Ronnie Cheung, Hassan B. Kazemian
London Metropolitan University,
United Kingdom
ccheung@acm.org, h.kazemian@londonmet.ac.uk

Abstract. The underlying technologies driving the World Wide Web are largely based on the assumption of wired communications and powerful desktop hardware. This is not true when a user is accessing the Web pages and documents using a PDA and moving from wireless network in a mall to a neighboring office environment. Taking into consideration on the information needs of the user, we have developed a meta-search application that can be executed on mobile devices with limited hardware capabilities. By running in the auto-configuration mode, the system detects the dynamic contextual information, and presents the results of meta-searching in a suitable format to cope with the limitations of the mobile stations. This is achieved by an adaptive service composition that is defined according to the adaptation model defined in a system configuration file.

1   Introduction

With the rapid development of the World Wide Web, many search engines are developed to facilitate information retrieval and filtering [1]. A meta-search engine is a kind of search engine that do not maintain its own databases, but search those of other engines [12]. Metacrawler, for instance, searches the databases of: Lycos, WebCrawler, Excite, Altavista, and Yahoo. It queries the other search engines, organizes the results in a uniform format, ranks them by relevance, and returns them to the user. In this paper, we describe the development of a meta-search engine that provides flexible and adaptive Web meta-searching service in a mobile environment. Compare with the fixed and wired network, the wireless mobile environment has smaller bandwidth, higher latency and higher error rates. A meta-search engine for mobile devices facilitates efficient web access by consolidating the search results from different search engines, and formats the results according to the requirements of mobile devices. A major issue is that many of the mobile devices have limited capabilities. For example, the size of screen, available memory and the depth of color are usually deficient. Conventional meta-search engines return the search results by using a single presentation format. In order to reduce the cost and increase the efficiency, and to provide a desirable Web meta-searching experience for the mobile

users, a Web meta-searching system for adaptive mobile environment is developed in this project.

The WebPADS system is employed as the underlying Web proxy framework to facilitate the adaptation requirements. Unlike other agent-based Web proxy systems, the WebPADS system offers a more flexible and adaptive platform for developing the meta-searching services. The details of the WebPADS platform are described in Section 2 of this paper. In the system, the client station acquires the contextual information by a service program running in the mobile node. The embedded service program then sends the updated information to the server periodically. Based on the dynamic contextual information of the mobile clients, the meta-search proxy server composes the service chains according to the dynamic requirements for adaptive service composition for meta-searching in a mobile environment. The client application is a user-friendly graphical user interface that is developed by Java technology. Users can interact with the system by submitting their specific requirements with a few steps. Most of the complicated tasks are processed in the proxy server. In the auto configuration mode, the users are only required to submit the keyword for the queries. The system can improve the efficiency and readability of the meta-searching, and the cost of meta-searching in mobile devices are decreased by minimizing the amount of data to be downloaded.

The major contribution of this paper is the abstraction of the dynamic service composition process using the WebPADS architecture. The integration of service composition is demonstrated using a meta-search engine application as an example. The proposed framework supports a self-organizing approach that allows a user to specify the desired adaptation and quality-of-service (QoS) requirements through adaptation rules. These QoS characteristics are broken down into requirements for individual services in the rule base using a novel approach for active service composition [9].

2   The WebPADS Platform

To provide and facilitate an effective Web meta-searching experience for mobile environments, the WebPADS system (**Web P**roxy for **A**ctively **D**eployable **S**ervice) has been employed as the underlying Web proxy framework for development. WebPADS is a object-oriented system based on the active service deployment architecture, dynamic service composition framework. Active service deployment and dynamic reconfiguration of mobile computing services accomplish the robust and efficient adaptation to mobile environment. To regulate the service configuration policies, WebPADS maintains a configuration description file within both of client and server sides. The configuration description files are in XML [3, 10] format and its acts as a declarative guide for dynamic service composition to adapt the vigorous changes of wireless environments. The condition of environment is reported by the client program and stored in an XML file in server side and its corresponding chain of service actions are defined in the configuration file.

We have previously described the WebPADS platform – a Web **P**roxy for **A**ctive **D**eployable **S**ervice [7, 8]. The WebPADS platform is an object-oriented system that

is based on active service deployment architecture, comprising components of dynamic service objects called *mobilets*, which executes under the WebPADS execution environment. Among the system's components, the event register allows objects to locate and register for event sources. When an event occurs, the objects that have registered for that event source are notified. Event sources include various changes in the status of a network, machine resources and connectivity. Furthermore, the composition of the services of the WebPADS server can be dynamically reconfigured to adapt to the vigorous changes in the characteristics of a wireless environment.
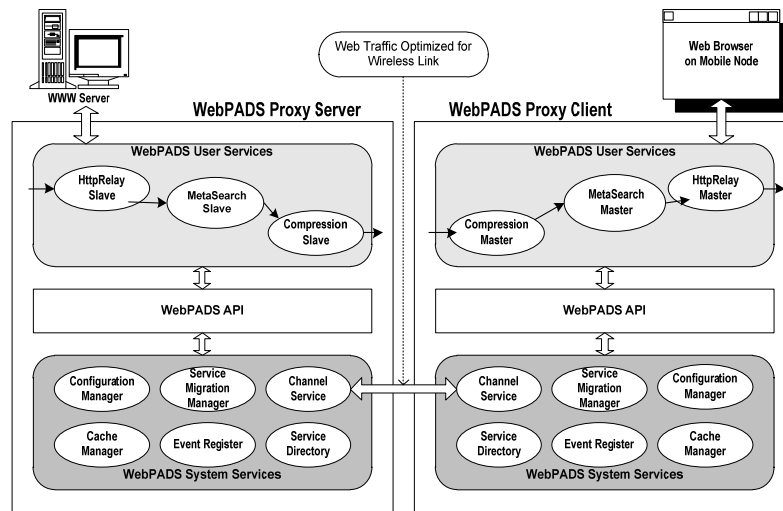

Fig 1. The WebPADS system architecture

The unit of service is implemented as a mobilet that provides value-added services to a wireless environment. The system's components also provide generic facilities that serve the mobilets, which in turn provide services that enhance a user's Web-browsing experience. A series of mobilets can be linked together to form a processing-service composition, allowing WebPADS to benefit from the aggregated functionality of a group of services (as shown in figure 1). Services provided by the WebPADS system can be migrated from one WebPADS server to another to track the mobility of a mobile node and serve the mobile node continuously. Services can also be added, removed and updated dynamically without interrupting the service provision to other mobile nodes. Furthermore, the composition of the services of the WebPADS server can be dynamically reconfigured to adapt to the vigorous changes in the characteristics of a wireless environment.

3   Systems Architecture

With the advent of mobile computing, it is common for users to access the Web through various mobile devices such as mobile phones, hand-held PCs and PDAs etc.

Each mobile device has different characteristics. For example, a user using a personal digital assistant may have less than 480 x 640 resolution and Web access through the mobile device is normally performed through a few proxy servers, either through a home wireless network or a wireless gateway installed in a mobile user's office. To provide adaptation to suit the requirements of mobile users, we have employed the WebPADS client-proxy mobile architecture to improve web searching experience. Figure 2 shows the system's architecture of the mobile meta-searching engine. Users can interact with the system using the client application in the mobile node. The Environment Monitor on the WebPADS client reports the contextual information of the mobile node, and the dynamic information of the mobile environment is updated to the server at regular intervals. The idea is to implement the WebPADS services both in the mobile node and in the server, where all HTTP requests or responses are intercepted, transformed and optimized for transformation over a wireless network. When a meta-search query is sent to the proxy server, a chain of service is created according to the dynamic configuration of the mobile clients, to facilitate mea-searching in a mobile environment.
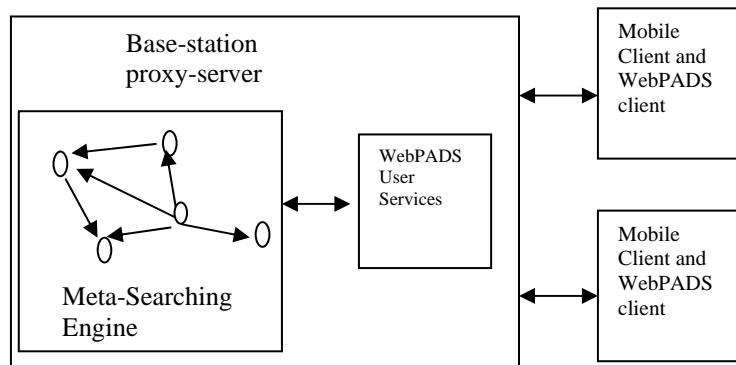

Fig. 2. System Architecture

4 Active Service Composition

Based on the user requirements of the dynamic mobile environment, the WebPADS system introduces an active service model that extends the client-proxy model to support active service composition. In this active service model, the proxy is composed of a service object called mobilets. This model offers flexibility because the chain of mobilets can be dynamically reconfigured to adapt to the vigorous changes in the characteristics of a wireless environment, without interrupting the service provision for other mobile nodes. Figure 3 shows the chain of mobilet services for providing adaptive meta-search services and Figure 4 shows the services for formatting search results.
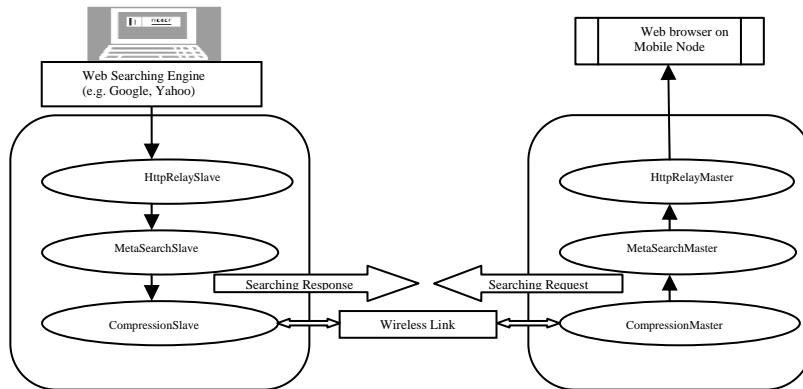
4

Fig. 3 The Service Chain for Meta-search

As shown in figure 3, for each service, the corresponding mobilets are implemented in pairs. For example: The HttpRelaySlave mobilet receives the http response from the Internet and the HttpRelayMaster mobilet displays the results in suitable format for the mobile station. The MetaSearchSlave mobilet searches the desired search result from the http response and the MetaSearchMaster mobilet converts the results from the meta-search engine for display. Finally the CompressionSlave mobilet compresses the retrieved results using the ZLIB library, and the CompressionMaster mobilet decompresses the retrieved result using the ZLIB for decompression. Each pair of mobilets in the service chains are dedicated for a specific function for dynamic service composition. Table 1 lists the functions of the corresponding mobilets within the service chains (as shown in Figure 4) for formatting the results of the meta-search engine.
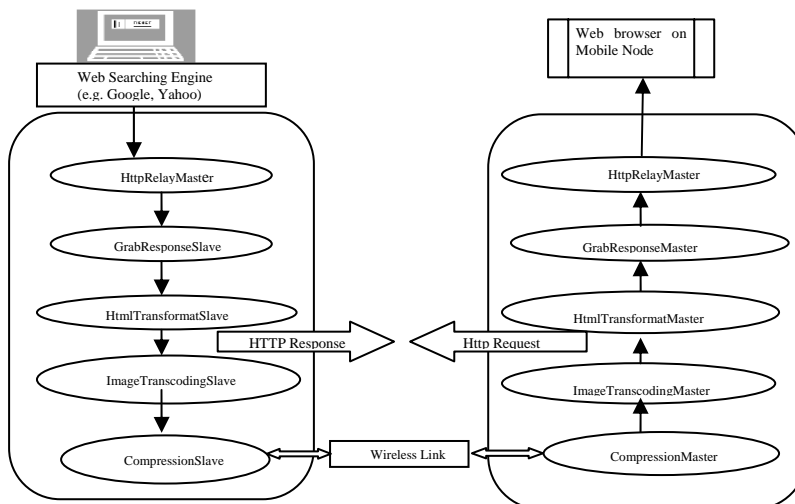

Fig. 4 Service Chain for Formatting Web Content

Table 1. Details of the services in the service chain for formatting the meta-search results

HttpRelaySlave	Receives the http response from the internet.
GrabResponseSlave	Retrieves all available image links from the html document, reformats the links with the domain and arrange the html and image link objects.
HTMLTransformatSlave	Reformats the html documents so that useless contents are removed. E.g. reformat the link of CSS, remove the javascript, applet and comments.
ImageTranscodingSlave	According to the conditions specified in the system configuration file, ImageTranscoding Slave can be used to transcode the image into a suitable format for the mobile client.
CompressionSlave	Compresses the retrieved result using the popular ZLIB compression library.
CompressionMaster	Decompress the retrieved result using the popular ZLIB compression library.
ImageTranscodingMaster	Stores the retrieved image in the client machine.
HTMLTransformatMaster	Stores the retrieved html documents in the client machine.
GrabResponseMaster	Commits the result storage and backup the old versions of the search result (if the same page is requested more than once).
HttpRelayMaster	Displays the results in a suitable format.

5 Adaptation Rules

The WebPADS platform has been used to implement mobile applications to provide efficient Web browsing experience for mobile users [4, 5, 6]. Table 2 lists the contextual information that is identified for the mobile devices. The information collected is used to configure the service provision dynamically to meet the processing and presentation requirements of the mobile clients. To regulate the dynamic service composition policies, the WebPADS system maintains a configuration description file utilizing XML. To dynamically adapt to the changes in the environment, WebPADS employs the environment monitor and event system to monitor and communicate the changes. An environment element consists of one or more conditions, where each condition specifies an event and a relational value that will fulfill that condition.

When a WebPADS client starts, a default service composition is created that is based on the description of the XML configuration file. At the same time, a number of alternative reconfiguration service chain maps are also created. Each map is attached to an environment monitor, which regulates the time and conditions for the service reconfiguration to take place. When all the conditions monitored by a specific environment monitor are fulfilled, the current service composition will be reconfigured to the service chain map attached to that environment monitor.

Table 2. Contextual information reported by the WebPADS client

Unit Type	Contextual Information
System	CPU type, clock rate, utilization, OS name, version
Storage	RAM size, free space, Secondary Storage size, free space
Network	Network type, capacity, data rate, delay, error rate
Power	Battery maximum lifetime, remaining lifetime
Display	Color type, depth, Screen resolution
Audio	Type of sound supported
Browser	Browser name, version

```xml
<RECONFIGURATION>
    ...
    <ENVIRONMENT config_option="COMPRESSION">
        <CONDITION_ATTRIBUTE value="CPU_USAGE">
            <CONDITION>
                <RELATION operator="LESS_THAN">50</RELATION>
                <CONFIG_VALUE>BEST_COMPRESSION</CONFIG_VALUE>
            </CONDITION>
            <CONDITION>
                <RELATION operator="MORE_OR_EQUAL">50</RELATION>
                <RELATION operator="LESS_THAN">65</RELATION>
                <CONFIG_VALUE>DEFAULT_COMPRESSION</CONFIG_VALUE>
            </CONDITION>
            <CONDITION>
                <RELATION operator="MORE_OR_EQUAL">65</RELATION>
                <RELATION operator="LESS_THAN">75</RELATION>
                <CONFIG_VALUE>LOW_COMPRESSION</CONFIG_VALUE>
            </CONDITION>
            <CONDITION>
                <RELATION operator="MORE_OR_EQUAL">75</RELATION>
                <CONFIG_VALUE>NO_COMPRESSION</CONFIG_VALUE>
            </CONDITION>
        </CONDITION_ATTRIBUTE>
    </ENVIRONMENT>
    ...
    <ENVIRONMENT config_option="IMAGE_SIZE">
        <CONDITION_ATTRIBUTE value="WIDTH_PIXELS">
            <CONDITION>
                <RELATION operator="LESS_OR_EQUAL">320</RELATION>
                <CONFIG_VALUE>1/2</CONFIG_VALUE>
            </CONDITION>
            <CONDITION>
                <RELATION operator="LESS_OR_EQUAL">480</RELATION>
                <RELATION operator="MORE_THAN">320</RELATION>
                <CONFIG_VALUE>2/3</CONFIG_VALUE>
            </CONDITION>
            <CONDITION>
                <RELATION operator="LESS_OR_EQUAL">640</RELATION>
                <RELATION operator="MORE_THAN">480</RELATION>
                <CONFIG_VALUE>3/4</CONFIG_VALUE>
            </CONDITION>
            <CONDITION>
                <RELATION operator="MORE_THAN">640</RELATION>
                <CONFIG_VALUE>1</CONFIG_VALUE>
            </CONDITION>
        </CONDITION_ATTRIBUTE>
    </ENVIRONMENT>
    ...
</RECONFIGURATION>
```

Fig 5. The XML Configuration File

The configuration file is used for constructing the user adaptation model. Figure 5 shows examples of adaptations rules that are used to compose the services for meta-search dynamically. It is implemented in an XML format file. For example, the rule segments corresponding to the adaptation rules for CPU usage can be interpreted as follows: If cpu_usage < 50 then perform Best_Compression; If cpu_usage > = 50 and cpu_usage < 65 then perform Default_Compression; If cpu_usage >=65 and cpu_usage < 75 then perform Low_Compression; If cpu_usage > = 75 then perform No_Compression.

6   Example – Dynamic Activation of Transcoding Service

The meta-searching system is implemented with a graphic user interface on the mobile client station. The current version supports meta-searching through Yahoo, Google, AltaVista and Lycos (as shown in Figure 6). Corresponding to the selected search engines and the selected configuration options (e.g. auto option for compression and image color level defined in the configuration file), the system communicates with the meta-search engine through the WebPADS server. The following example shows the results of adaptive service composition after the dynamic activation of the image transcoding service in a low bandwidth wireless network environment (as shown in figure 6 and figure 7).
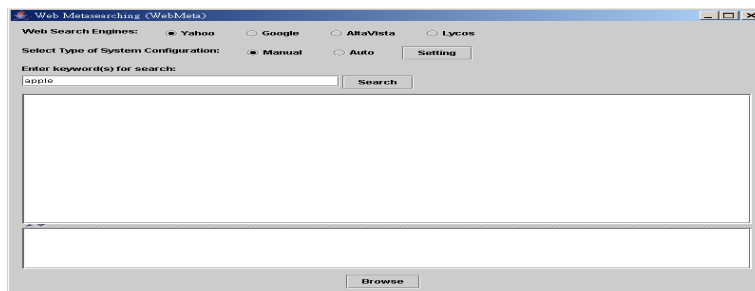

Fig 6. The graphical user interface in the mobile station

In the meta-searching system, the size of images within the resulting pages directly affects the amount of data that is transported in the mobile environment. For example, if the bandwidth of the mobile connection is less than a threshold value (e.g. 64 K), it is desirable to transcode the graphic files (e.g. the GIF files) with a lower bit depth to reduce the image file size [2, 11]. The rules for adaptations can be configured in the configuration file in WebPADS. According to the bandwidth and contextual parameters, a scaling parameter can be defined to determine how much an image is downsampled. According to the dynamic adaptation requirements, the number of colors in a color-mapped image can be reduced (e.g. a 24-bit color image may be converted to 8-bit grayscale or even monochrome representation). Figure 7 shows the examples of the original graphic image and the retrieved image (down-sampled) after dynamic activation of the transcoding service in the system.
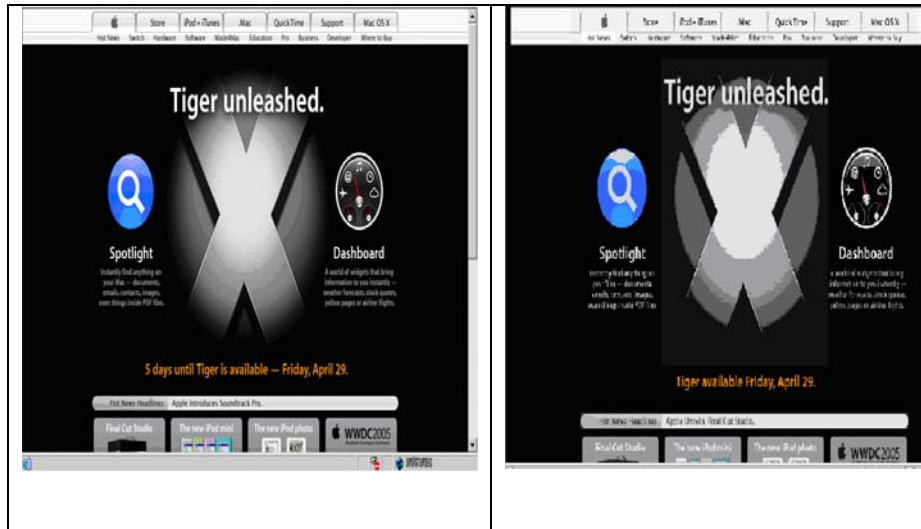
8

Fig 7. The original graphic image (left) and the down-sampled image (right) in the mobile station

7 Conclusion

A meta-search application is developed for mobile devices with limited hardware capabilities. By running in the auto-configuration mode, the system detects the dynamic contextual information, and presents the results of meta-searching in a suitable format to cope with the limitations of the mobile stations. This is achieved by an adaptive service composition that is defined according to the adaptation model defined in a system configuration files. The configuration file can be stored at the mobile device and edited by the users at any time. The parameters for the adaption model can be adjusted to suit the information requirements of the mobile users. By implementing the system in the WebPADS platform, it is possible to maintain on-going service provision as the mobile node moves across different domains (e.g. from a home network to an office network in a wireless environment). A major concern about our implementation is that it is not easy to maintain the adaptation rules stored in the system confirmation file in XML format. Future directions of research include using artificial intelligence techniques to improve the flexibility in adaptation, and to provide user-friendly interfaces for the users to maintain the adaptation rules in the rule base.

References

1. Belkin N., Croft, W. B.: Information Filtering and Information Retrieval: Two Sides of the Same Coin? Communications of the ACM, 35(12), pp. 39-38 (1992)

2. Bharadvaj, H., Joshi, A., Auephanwiriyakul, S.: An Active Transcoding Proxy to Support Mobile Web Access. In: Proceedings of the 17th IEEE Symposium on Reliable Distributed Systems, pp. 118-123 (1998)

3. Bradley, N.: The XML Companion. Addison Wesley (1998)

4. Cheung, R.: An Adaptation Control Model to Support Mobile Web Access. International Journal of Control and Automation, 1(1), 9-16 (2008)

5. Cheung R.: A Context-Aware Adaptation Model for Efficient Web Browsing on Mobile Devices. Advances in Communication and Networking, Communications in Computer and Information Science, vol. 27, pp. 41-54, Springer (2009)

6. Cheung R.: An Adaptation Model for Accessing Document Clusters in a Mobile Environment. In: Proceedings of the 8th Asia-Pacific Conference on Computer-Human Interaction, Seoul, Korea, pp. 5-8 (2008)

7. Chuang, S. N., Chan, A. T. S., Cao, J., Cheung, R.: Dynamic Service Reconfiguration for Wireless Web Access. In: Proceedings of the Twelve International World Wide Web Conference, pp. 58-67, ACM Press, Budapest, Hungary (2003)

8. Chuang, S.N., Chan, A.T.S., Cao, J., Cheung, R.: Actively Deployable Mobile Services for Adaptive Web Access. IEEE Internet Computing, Mar/Apr, 26-33 (2004)

9. Binder, W., Bonetta, D., Pautasso, C., Peternier, A., Milano, D., Schuldt, H., Stojni´c, N., Faltings, B., Trummer I.: Towards Self-Organizing Service-Oriented Architectures. In: Proceedings of the (SEASS 2011) IEEE Fifth International Workshop on Software Engineering for Adaptive Service-oriented Systems, in press, Washington, USA (2011)

10. Extensible Markup Language (XML) http://www.w3.org/XML/

11. Han, R., Bhagwat, P., Lamaire, R., Mummert, T., Perret, V., Rubas, J.: Dynamic Adaptation in an Image Transcoding Proxy for Mobile Web Browsing. IEEE Personal Communications, 5(6), 8-17 (1998)

12. Zhao, D. J., Lee, D. L., Luo, Q.: A Meta-search Method with Clustering and Term Correlation. In: Proceedings of the International Conference on Database for Advanced Applications: pp. 543-553, South Korea, (2004)