# Small Devices and the Wireless Internet

Nagashree Suryanarayan
Graduate student, Dept of Computer Science
University of Texas at Arlington
suryanar@cse.uta.edu

## ABSTRACT

The next chapter in extending the Internet and the integration of wireless devices into our everyday lives is being written right at this very moment. This paper is a study of the current state of technology and a look at the future. The study looks into the areas of effective User Interfaces, efficient bandwidth utilization and improved latency in small wireless devices. "Digestor" and "m-links" are proxy-based systems that achieve better user interfaces in small devices for displaying the contents of the World Wide Web. "WebPADS" and "Pythia" are proxy-based systems for low bandwidth surfing in a high bandwidth world. "TranSquid" is an intermediary caching mechanism based on device profiles to improve availability. The paper also looks at various wireless sensors, which are replacing mechanical transducers and help to achieve virtual presence.

## INTRODUCTION

"Information Appliances" – pervasive devices such as cell phones and Personal Digital Assistants, PDAs that offer Internet connectivity have become a part of everyday business. This is driving the wireless world closer to the Information highway – the Internet. But the technology that drives it is still debated. Two divergent views are the Wireless Application Protocol, WAP approach from the mobile telephone industry, and the Wireless Internet approach from the computer industry. In the WAP approach, mobile telephone handsets have the computing capacity added, to become wireless data devices. Whereas in the Wireless Internet approach, hand held computers have wireless communications added to become multifunctional phones. This paper concentrates on the Wireless Internet approach for small devices.

In today's Internet world, most pages on the World Wide Web are designed for display on desktop computers with color monitors. However, hand held devices are inherently limited by their small screen size, limited keyboard than fixed units. This can lead to 4:1 or even greater ratio of designed versus available screen area, making direct presentation of most WWW pages on these small devices aesthetically unpleasant, un-navigable, and sometimes completely illegible. Accessibility features can be added most easily when the web content is created. Simplest example is alternative text for images. But this seems infeasible for the millions of pages already present. To make it better for the small device users, a number of schemes have been suggested to improve display and navigation. "Digestor"[4], a software which automatically re-authors arbitrary documents from the World Wide Web to display appropriately on small screen devices, providing device independent access to the web. Digestor is implemented as an HTTP proxy which dynamically re-authors requested web pages using a heuristic planning algorithm and a set of structural page transformations to achieve the best looking document for a given display size. Another ideology for fitting Desktop contents to a small display calls for separate interfaces for navigation and content manipulation. "m-links"[5], a middleware proxy system supports this dual mode browsing, offering mobile users a range of actions on any web link.

The wireless environment is drastically different from the wired counterpart. In particular, bandwidth, latency and error rate are much poorer, varying and asymmetric in nature, which lead to sub-optimal utilization of the wireless link when used by Web applications. A web-proxy architecture called "WebPADS"[3] is developed to enhance Web applications running on wireless network. WebPADS provides mechanisms that automatically locate and configure a flexible and adaptive wireless Web proxy. "Pythia"[9], a HTTP proxy based on real-time distillation provides the user with

a powerful mechanism for management of limited bandwidth, without completely sacrificing bandwidth-intensive non-textual content. For large images, a scaled down image is loaded initially with a provision for refinement, if necessary. Thereby reducing the large initial latency considerably.

The client space has become heterogeneous in terms of device capabilities. Small devices are ubiquitous and form a significant part of the Internet client community. Transcoding has been a popular technique to render data for small devices that have smaller displays, and lesser color capabilities. But transcoding comes at the cost of caching at the Intermediary. "TransSquid"[8], a transcoding and caching proxy that caches objects for heterogeneous client spaces by maintaining separate caches for different categories of clients, classified by their device profiles.

The future looks bright for Smart sensors essential for monitoring and diagnosing systems remotely. These sensors can be monitored and controlled through the wireless Internet and are all pervasive in the health care industry. They are even extendible to centralized control by remote sensing in the defense industry, thus enabling Virtual Presence [12].

## USER INTERFACE

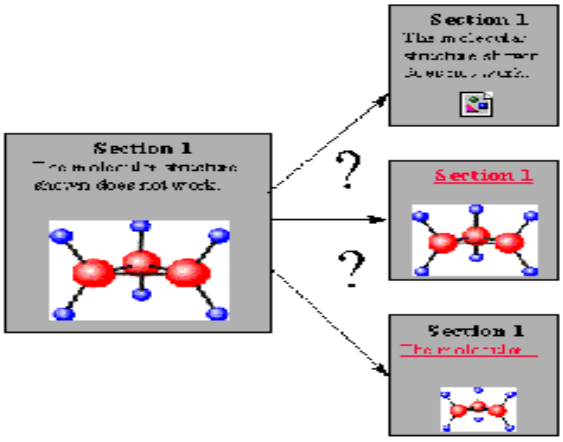
Simply designed web pages will display well on hand held devices. The text can be enlarged for reading on the small screen, images can be omitted or selectively downloaded as required. However this requires the designer to be familiar with accessibility issues and incorporate those on the web page design.

### Device Independent access

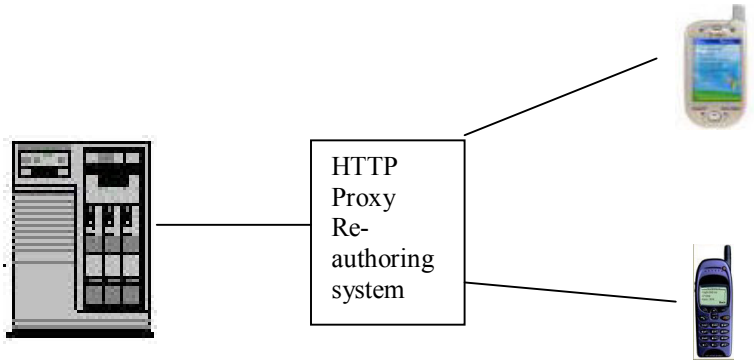
The four general approaches to display WWW pages on small screen devices are device-specific authoring, multiple-device authoring, client-side navigation and automatic re-authoring. Device-specific authoring involves authoring a set of WWW pages for a particular display device. The users of such specialty devices will only have access to a select set of services, and the pages for these services can all be designed up-front for the device's particular display. Multiple-device authoring identifies a range of target devices and mappings from a single source document to a set of rendered documents are defined to cover the devices within the range. Client-side navigation provides the user with the ability to interactively navigate a single web page by altering the portion of it that is displayed at any given time. An example of this is the use of scroll bars on the document display area. Automatic re-authoring involves developing software which can take an arbitrary web document designed for the desktop, along with characteristics of the target display device, and re-author the document through a series of transformations so that it can be appropriately displayed on the device. This process can be performed either on the client, on the server, or on an intermediary HTTP proxy server.

### Digestor System

The Digestor system explained in [4] has two major elements. A collection of individual re-authoring techniques which transform documents in various ways; and an automated re-authoring system which implements a design strategy by selecting the best combination of techniques for a given document/display size pair. The Section header outlining technique is one of the re-authoring system where each section is deleted from the document and the section header is converted into a hypertext link which, when selected, loads the deleted content into the browser. Another technique involves replacing each block by its first sentence and a link to the block. The conversion is shown in figure below.




**Architecture**




The Digestor system is implemented as an HTTP proxy server. The block diagram is shown in the fig. The first thing that a user of Digestor will typically do is specify the size of display for their device, and indicate the size of their default browser font. The proxy accepts a request for an HTML document, retrieves the document from the specified HTTP server, parses the HTML and constructs an Abstract Syntax Tree (AST), labels each of the AST nodes with a unique identifier and then retrieves any embedded images so that their size can be determined. Once this has been accomplished, the planner is initialized with a state containing the AST for the original document. During each planning cycle, the planner selects the state with the best document version so far, then selects the best applicable transformation technique and applies it resulting in a new state and document version being generated. Fifteen transformation techniques have been implemented and integrated into the system. Transformations manipulate the AST in the state they are applied to in order to produce a new version of the document. Whenever portions of the AST are deleted or transformed, an HTML hypertext link is added into the AST referencing the node identifiers of all affected AST subtrees, enabling users to request the original portions of the document that have been modified during re-authoring.

**Desktop to Phonetop**

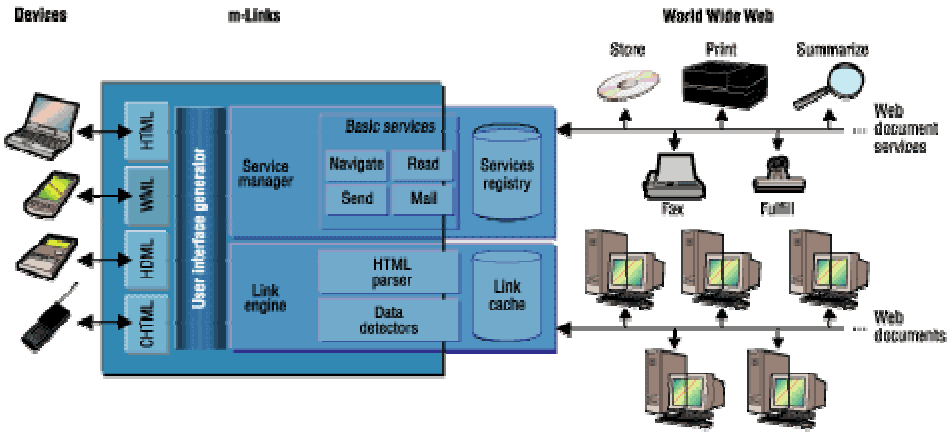
Mobile Web devices span a range of capabilities. The "Fitting desktop content in a small display" sidebar describes fitting techniques in transducing and three other categories: scaling, manual authoring and transforming, in paper [ 5]. Transforming is similar to professional content tailoring to

a particular device yet without the manual overhead. The Digestor system described above is an example of such a transforming process. But for very small devices, the interface structure was just too difficult for users to understand and navigate. A new browsing model that supports navigation and action in separate interfaces is described in paper [5].

**m-links**

m-links is a middleware proxy system that retrieves Web documents using HTTP, lets users navigate and apply services to Web content, and delivers a suitable interface to a variety of small Web-capable wireless devices. M-links is both simple to use and powerful. Separating links from page content makes navigation a matter of selecting a link form the list. Because the users can apply various Web-based services such as print, fax etc they can do more with content than simply read it on their phones. A Web page opened through the m-links proxy consists of a list of links from that page and can dig through the list in the same way they dig through folders in a file dialog to select a filename. They may invoke a service when the appropriate link is found. Consider a simple example where a user navigates to a link of a PDF document and e-mails the PDF to himself for later use at the desktop by selecting the link and applying the e-mail service. A link is the basic unit of manipulation, but sometimes information the user wants may not be explicitly linked in HTML. m-links solves this by scanning the text, using server-side data detectors to create new links, and then including these links in the list of links that can receive some user action.

**Architecture**




The figure depicts the m-links architecture. m-links has three main parts namely, the link engine, the service manager, and the user interface generator. The link engine processes Web pages into a collection of links that is stored in the link cache. A request to the navigation interface for a Web page involves –

1. Page parsing – creates a parse tree that consists of the HTML tags and text on the Web page.
2. Data detection – invokes server-side data detectors to identify Web page elements that are not HTML anchors or tags but that could receive some user action.
3. Link naming – uses a link naming algorithm to determine concise but meaningful link labels
4. Link categorization – hierarchically categorizes links

The service manager creates the action interface, which presents a menu of tools that users can invoke once they have located a link with the navigation interface. Once the user selects a link in the navigation interface and chooses tools, m-links receives the request and passes the link to the service manager. The service manager constructs the action list by evaluating a set of rules that each service specifies against the link's attributes and the characteristics of the user's device. If the service

manager determines that the service satisfies all the service rules, it adds the service to the action interface. The User interface generator uses a combination of screen template substitution and program inheritance to produce the appropriate mark-up interface for each device.

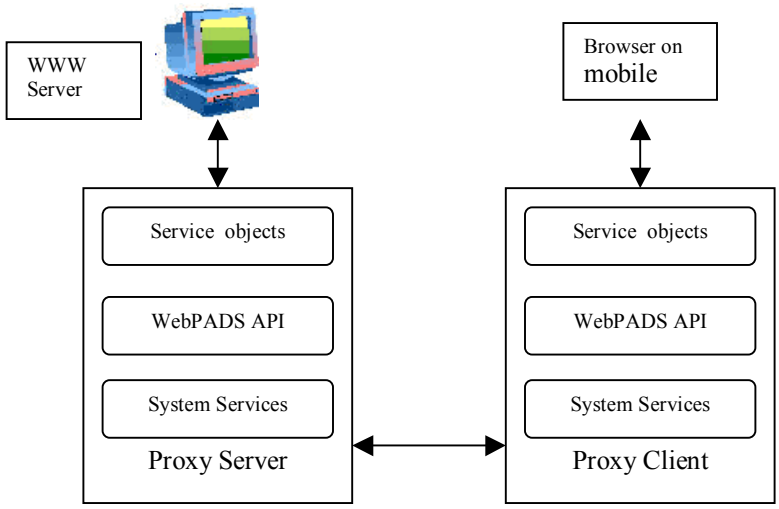
## BANDWIDTH UTILIZATION

### WebPADS

Web Proxy for Actively Deployable Services, WebPADS provides mechanisms that automatically locate and configure a flexible and adaptive wireless Web proxy described in paper [3] to overcome the bandwidth constraints. This is based on the active service deployment approach. In active service deployment model, the proxy is composed of a chain of service objects. The service objects can be dynamically reconfigured to adapt to the vigorous change of characteristics of wireless environment, without interrupting the service provision for other mobile nodes. Also, the service objects can be migrated to a new proxy server when the mobile node moves to a new network.

Consider a scenario when a user uses his PDA to access the web in his office using wireless LAN. The PDA is installed with the WebPADS proxy client (WpsClient), which acts as a proxy server to the browser. On the other side of the fixed network, a WebPADS proxy server (*WpsServer*) is running in complement to the W*psClient*. When he starts the *WpsClient*, two services were pushed onto the *WpsServer* and executed on both the *WpsClient* and *WpsServer*. Each service is comprised of both the master and slave services, which are executed as peer-to-peer objects running on the *WpsServer* and *WpsClient*, respectively. A slave page reformatting service object on the *WpsServer* is responsible to reformat all requested Web pages to fit the screen size of the PDA, while a master page reformatting service object running on the *WpsClient* to control the parameters of the slave service. Also, a slave compression service is running on the *WpsServer* to compress the incoming response before sending to the *WpsClient*, in order to minimize the bandwidth consumption and speedup the transmission. A corresponding master compression service is also running on the *WpsClient* to decompress the incoming response. When the network bandwidth of the PDA drops to a faction of the original capacity. The *WpsClient* detect the decrease of bandwidth and starts a image transcoding service to further speedup the response time. A slave image transcoding service is pushed onto the WpsServer to scale down the dimension of images, increase the compression ratio of JPEG images. The *WpsClient* connects itself to the new *WpsServer*, and instructs the new *WpsServer* to carry out a migration of services from the old *WpsServer*. As all the states of the services are maintained in the object migration process, the *WpsClient* can continue the previous web session without major interruption.

### Architecture

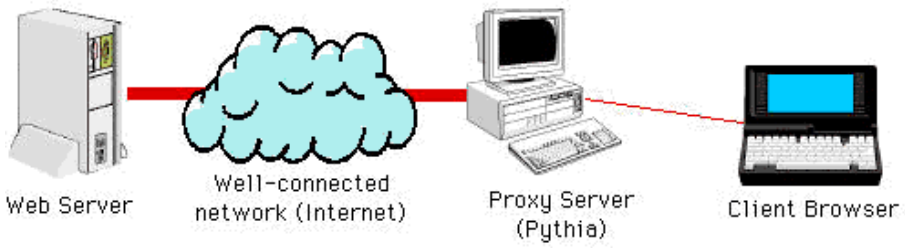


WebPADS adopts a layered approach. Each service provides a single functionality and is comprised of a master service object and a corresponding slave service object that are located on the proxy client and the proxy server respectively. It is possible for mobile client to carry with it relevant services as it travels across foreign domains. As the need arises, service objects from the client can be dynamically pushed to the proxy and configured to operate in a coordinated manner. Conversely, it is possible to for proxy to push service objects to the client. Service objects exist in pairs, a master object that resides at the *WpsClient* and a slave object that resides at the *WpsServer*. The slave object shares a major portion of the processing burden, while the master object instructs the slave object what actions should be carried out and presents the processed output to the *WpsClient*. System services provide essential services for the WpsClient and WpsSerevr to cooperate with each other.

### Pythia

The loading of Web pages take a painstakingly long time on small wireless devices. Progressive and interlaced GIF and JPEG display a blurry image initially and refine it as more image data arrives. However, for large images, the initial latency is still high. Pythia, a HTTP proxy based on Real-time distillation described in paper [9] is a feasible solution. It comprises of a distillation and a refinement process. Distillation is highly lossy, real-time, datatype-specific compression that preserves most of the semantic content of a document. Refinement is a process of fetching some part (possibly all) of a source document at increased quality, when required. Because distillation can be performed in real time, it eliminates the need for servers to maintain multiple intermediate-quality representations of a document: Any desired intermediate representation can be created on demand using an appropriate distiller. Distillation and refinement allow us to trade cycles, which are cheap and plentiful, for bandwidth, which is expensive and scarce. Intelligent distillation will scale the source image down to reasonable dimensions for the client display, and preserve only the color information that the client can display. Distillation thus allows bandwidth to be managed in a way that exploits the client's strengths and limitations.

### Architecture




The fig depicts the architecture of Pythia. Rather than fetching a URL directly from the appropriate server, the fetch request is passed on to the proxy. The proxy obtains the document from the server on the client's behalf, and forwards it to the client. Pythia can run anywhere inside the wired Internet.

## DEVICE PROFILING

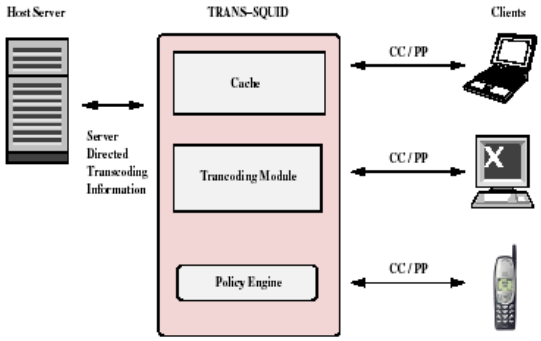
### TransSquid

Heterogeneous client space exists when data is available to users across a growing number of destinations - laptops, desktops, kiosks, automobile browsers, cellular phones, pagers, pocket PCs, Palm OS devices and other handheld devices. Caching is a proven intermediary process that

improves user satisfaction by serving web objects locally from a proxy that is near to the client. One of the fundamental problems that appear in enabling caching for heterogeneous environments is storing of multiple variants of the same object generated because of the transcoding operation at the server side or at an intermediary before the caching server. Presently, all transcoding engines mark transcoded content as un-cacheable. This is unnecessary and leads to wastage of bandwidth and increases latency in a response that needs transcoded data since the content now travels across the Internet when it could very well be served from a location near the client. Paper [8] presents a novel caching and transcoding system called *TransSquid*. *It* is a modular framework that enables caching in heterogeneous environments by maintaining a multi-level cache and linking it with the transcoder.

The content that needs to be sent to the client is based on the client device profile provided during to content negotiation. Transcoding and content negotiation lead to multiple variants of a resource or a web object that differ in modality and fidelity. For example, a typical PDA client would be satisfied with a lesser fidelity variant than the variant for the same resource for a Workstation client that would demand rich features. Hence, intelligent caching policies are required to avoid redundancy when different variants of the same resource are to be maintained. It requires the understanding of the Clients Capabilities and Preference Profiles (CC/PP) [6] to facilitate client recognition. Intelligent caching proxies in a heterogeneous client environment could have the functionality to exploit the heterogeneity through the conversion of high fidelity variants of a resource to low fidelity variant to serve clients at the lower capacities. A typical example of this would be the conversion of high fidelity JPEG image to a low fidelity GIF image in reduced color and half the dimensions to serve a low-resolution PDA client.

**Architecture**




TransSquid architecture shown in the fig is designed as a smart intermediary that can cache and transcode web objects in an environment where the client requests have to be serviced intelligently according to the client capabilities. TransSquid tries to solve the problem of caching in a heterogeneous client environment by taking a client centric approach for categorizing different variants and then storing them in a multilevel cache on the basis of fidelity and the modality. TransSquid, therefore provides a limited level caching architecture, by dividing the client space into a limited number of (say three) categories based on the capabilities. A client device is a member of one and only one of the 3 categories depending on its capabilities like display size, colors, storage and bandwidth of connection.

High capability clients – Workstations and Laptops
Medium capability clients – PDAs and WebTops
Limited capability clients – Mobile phones
Thus a request would result in the retrieval of the object in the cache belonging to that class.

## SMART SENSORS

Smart sensors overcome the limitations of hard-wired transducers by demonstrating portability, extreme accuracy and easy control.

### LifeMinder

LifeMinder, a wearable healthcare support system, which consists of a wristwatch shaped wearable sensor module and a PDA. The wearable sensor module equipped with sensors of accelerometer, thermometer and a bluetooth module to communicate with the PDA. It monitors the users health condition, movements and behavior. The wearable sensor module measures the users physiological data and communicates with the PDA via a bluetooth module in order to dispense with complicated wiring between them. The PDA communicates with the server via the wireless Internet and sends the patient's information. The server evaluates the information and sends the user suitable healthcare instructions. In this way the system provides the user with real-time healthcare information.

### Sensors in Civil Infrastructure systems

There is a paradigm shift from wired transducers to wireless sensors. It is possible to build a wireless sensor that will record and transmit strain measurements for a period of over 10 years. These miniature wireless sensors are inserted in a concrete pour during construction and data can be received from these sensors via radio frequency.

### Automobile sensors

A wireless sensor can be place on the brakes of a vehicle that will convert the heat energy from the brakes into sufficient electrical energy to power measurement of the brake condition and the radio frequency transmission of this data to an onboard computer in the vehicle.

### Virtual presence system

The virtual presence system is based in a naval ship and employed with wireless technology maintaining the system. It consists of a set of wireless sensors, access points communicating with the central Watchstation via wireless LAN. The various sensors placed at different parts of the ship communicate with the access points, employing low power radio transmissions with IEEE 802.11 being employed. The access points are the interface between the sensor units and the ship's backbone LAN. The access points communicate with watch stations via the LAN. Each access point incorporates a camera and microphone providing virtual presence. This system aims at reducing the crew required on each ship by centralized control.

## DISCUSSION

Small devices are ubiquitous. This has forced researchers towards techniques for providing better Internet access from these devices. The three categories discussed involve better User interface, better Bandwidth utilization and accessibility based on Device profiling.

Device-specific authoring will typically yield the best-looking results, but limits the user's access to a small select set of web pages. Multiple-device authoring, while less total effort per document than device-specific authoring, still requires significantly more manual design work than simply authoring for a single desktop platform. Client-side navigation holds promise if a good set of techniques can be developed, but the 'peephole' approach taken in PAD++ seems very awkward to

use for large documents, and the active outlining technique has limited applicability since most web pages do not use a strict section/sub-section organization. Automatic re-authoring is thus the ideal approach to providing broad access to the web from a wide range of devices, if it can be made to produce legible, navigable and aesthetically pleasing re-authored documents without loss of information. m-links best supports directed browsing in which the user follows links in an attempt to find specific content. Portals like Yahoo can have hundreds of links, which makes that finding an appropriate link take a long time. Digestor performs better than m-links because this system separates page areas into distinct units for presentation.

WebPADS is built on the active service deployment model. It intercepts Web traffics at both the ends of the wireless link. Hence transformation can be more efficient and optimized based on the bandwidth availability. WebPADS can be migrated from one WpsServer to another to serve the mobile node continuously. Furthermore, the constitution of services of any user can be dynamically reconfigured to adapt the vigorous change of characteristics of the wireless environment. Pythia allows the user to bound latency and exercise explicit control over bandwidth that may be scarce and expensive. Knowledge of client display constraints allows content to be optimized for rendering on the client. But it cannot hide the server-to-proxy latency.

The device profile based approach in TransSquid is successful in improving latency because it caches transcoded versions of the Web objects, which typically are marked non-cacheable in the normal flat architecture. Also, CC/PP helps content negotiation in advance.

Another line of advance is the wireless sensors. In addition to the wireless capabilities for sensors, advances in microelectronics have also made possible complex data processing and reduction within the miniaturized confines of the sensor package. Thus sensors can be remotely programmed with data processing algorithms that can capture the information content of the sensor data without all the redundancy from mechanical transducers. The Virtual Presence system is a typical example of enabling centralized control through a wireless LAN, thereby reducing the manpower requirements.

## CONCLUSION

A new class of electronic devices with Internet access capability called "Information Appliances" is born. The devices such as Personal Digital Assistants (PDAs), car navigation systems and cellular phones belong to this category. Although the users of very small Internet devices will benefit when Web content is designed specifically for their devices, it seems unlikely that content providers will be able to customize all new content, let alone existing documents. Indeed, Web access for these users will undoubtedly become a more pressing concern as ever-smaller Internet devices become more pervasive. The various proxies discussed in the paper provide mechanisms that overcome one or more of the access constraints. These are the independent research units today that need better consolidation for tomorrows easy access from small devices.

## REFERENCES


1. Bill N. Schilit, Jonathan Trevor, David M. Hilbert, Tzu Khiau Koh "Web Interaction using very small Internet devices" 0018-9162/02 Cover Feature, IEEE 2002

2. Tom Worthington "Issues in the Wireless Internet" http://www.tomw.net.au

3. Siu-Nam Chuang, Alvin T.S. Chan, Jiannong Cao "An active service framework supporting wireless Web access" 0-7803-7376-6/02, IEEE 2002

4. Timothy W. Bickmore, Bill N. Schilit "Digestor: Device independent access to the world wide web" Proceedings of the sixth international World Wide Web conference, Santaclara, 1997

5. H.Rao, Y.Chen, D.Chang, M.Chen "I-mobile: A proxy based platform for mobile services" Proceedings of the first ACM WMI 2001, Rome

6. "Composite capability/ Preference profiles – a user side framework for content negotiation" W3C note, www.w3c.org/TR/Note – CC/PP

7. Cynthia Traeger "Wireless Security developments", Faulkner Information system

8. A. Maheshwari, A.Sharma, K. Ramamritham, P. Shenoy "TranSquid: Transcoding and caching proxy for heterogenous e-commerce environments" Proceedings of the 12th International Workshop on e-commerce, IEEE 2002

9. Armando Fox, Eric Brewer "Reducing WWW latency and Bandwidth requirements by Real time distillation" Proceedings of the fifth International World Wide Web Conference, Paris 1996

10. K. Ouchi, T. Suzuli, M. Doi "LifeMinder: A wearable healthcare system using User's context" Proceedings of the 22nd International Conference on Distributed Computing systems, IEEE 2002

11. Karl Kiefer "The slow evolution of the Wireless sensor" 0-7803-7231 IEEE 2002

12. Gary Schwartz "Reliability and Survivability in the reduces Ship's crew by Virtual presence systems" Proceedings of the International conference on Dependable systems and networks, IEEE 2002