A Data Repository for Fine-Grained Adaptation in Heterogeneous Environments

Calicrates Policroniades
Computer Laboratory
University of Cambridge
United Kingdom
cbp25@cl.cam.ac.uk

Rajiv Chakravorty
Computer Laboratory
University of Cambridge
United Kingdom
rc277@cl.cam.ac.uk

Pablo Vidales
Laboratory for Communication Engineering
University of Cambridge
United Kingdom
pav25@eng.cam.ac.uk

ABSTRACT

In this paper, we present DATOM – an extended structured data storage system for mobile data management. We propose co-operation of networks and applications in order to adapt to the limitations imposed by heterogeneous environments. We show how adaptation can be achieved to address the high variability in link-layer characteristics as typically seen in hybrid wireless networks.

We propose a novel data model for DATOM, that departs from the traditional view of files as monolithic objects, and breaks them into *elements* to enable fine-grained adaptation. We incorporate metadata such as access patterns and priority to individual elements in our data model. Content adaptation of individual elements is controlled by application-specific policies that are applied according to present and future network conditions. Using an integrated LAN-WLAN-GPRS testbed, we propose to evaluate a set of adaptation strategies for mobile data management that can help in improving the overall user experience.

Categories and Subject Descriptors

E.5 [**Files**]: Structure; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*retrieval models*

General Terms

Design

Keywords

mobile computing, data adaptation, overlay networks

1. INTRODUCTION

Data transmission adaptation in heterogeneous environments is a fundamental attribute that mobile systems should support. The need to transmit only an adequate amount of data in a suitable format at the correct time involves physical, operational, and economic factors. As a consequence, different adaptation strategies have been researched, for example: *network-adaptation* [1, 2], adaptation in *content transmission* [3, 4], and *application-specific* adaptation strategies [5, 6]. We consider that a practical solution should integrate these strategies and be able to efficiently cope with the changing conditions of heterogeneous environments.

File systems were never originally designed to address mobility as a critical design concern. As a result, previous proposals to adaptation in mobile environments have been built based on the traditional concept of a file as a monolithic object [7]. Solutions have been designed for specific wireless links [1, 2], and do not react accurately to heterogeneity (i.e. WLANs, 3G, GPRS). Consider for example a scenario in which a mobile user accesses a file with rich content (i.e. formated text, images, sound, or video) using a PDA, then he migrates from a WLAN link to a GPRS connection with wide-differences in link characteristics such as limited bandwidth, increased costs, and different traffic characteristics. Undoubtedly, it would be very useful to know about file's structure and nature of its content in order to efficiently adapt data transmission of individual parts of the file in order to maximise user experience and resource utilisation. Thus, manipulating files as whole objects can pose a serious impediment to efficiently adapt data for mobile users.

Content delivery can be improved significantly if it is assisted by the storage system in two ways. Firstly, with a storage abstraction in which files' logical *structure* can be explicitly exposed. This enables optimisations based on transmission of subsets of the data contained in the original file. Secondly, with a *richer metadata model* associated with the content of each individual data element. Such metadata can be access patterns, type, or priority. These features, unfortunately, are not available in the traditional file system paradigm.

As the environment is heterogeneous, the underlying network can change with time. A comprehensive system should detect changes and adapt data transmission according to the context. It should also anticipate network changes and inter-network handoffs, which is of significant advantage to mobile applications.

DATOM examines data transmission adaptation from two different angles. Firstly, it supports *content transformation* and *application-specific adaptation* sustained by a novel data model that certainly augments the storage system capabilities to adequately serve data to mobile applications. It defines *policies* for content transmission, quality, and extent of the information to be fetched from the data server. Secondly, it provides *network-specific* adaptation strategies to cope with drastic changes in link-conditions of different wireless networks at run time. The contribution of this work is to highlight the importance of fine-grained adaptation when shipping data over heterogeneous wireless environments.

The paper is organised as follows. Section 2 elaborates extensively on the overall design of DATOM. Section 3 describes our tightly-integrated LAN-GPRS-WLAN experimental testbed and our initial data transmission adaptation experiences. Section 4 focusses on our research directions, while Section 5 comments related work. Finally, section 6 concludes the paper.

MobiDE'03, September 19, 2003, San Diego, California, USA

2. DATOM SYSTEM

2.1 Design Rationale

DATOM believes in imposing certain degree of structure on data at the storage system level. The main idea is to work towards a storage system where a more granular and intelligent manipulation of data is feasible. By using this structural information, it is possible to incorporate useful fine-grained adaptation strategies. Consequently, files are not seen as monolithic entities of storage anymore, but as a *composite entity* – an aggregate of *data elements*.

DATOM presents a model for adapting applications to different service requirements and underlying networks. We argue that applications need to be adapted differently – editing tools (poor in graphical content and usually long connections), web browsing (rich graphical content and typically short-lived connections), and video streaming applications (rich content and real-time demands).

Applications usually implement adaptation relying on their own set of libraries. Firstly, to extract any structure from their data files, and, secondly, to define basic transformations on them. For example, an ELF file may contain different object files. All these elements are *not naturally exposed and manipulated* by the file system. Web servers parse HTML files in order to disclose offline elements such as images. Applications are only left with the option of building their own routines to extract and manipulate these individual data elements. If a user is interested in a specific element inside the file, he would most likely ship the whole data set [6][1]. Consider a user downloading a whole paper in PDF format only to realise that it is not relevant. Thus, in a mobile environment it is desirable to ship only the most important elements or those that will be actually useful according to current user intents and context. Adaptation infrastructure that storage systems provide to applications is minimal, if not inexistent.

With DATOM, we aim to provide application developers with the facilities to naturally create different adaptation strategies on top of the storage system. DATOM enables fine-grained manipulation of data, minimises user-perceived latencies and communication disruptions, while maximising bandwidth utilisation.

2.2 The Data Shipping Process

DATOM's key components are the *data server* and a *client middleware* on the mobile host that acts on behalf of the application. Figure 1 depicts the main interactions of the system when serving data to applications. Initially, the application communicates its interest on certain information to the client middleware. The client middleware requests this information from the data server, which replies with only a compositional *sketch* of the information; no content is transfered at this point. The client middleware then passes a copy of this sketch to the application, while simultaneously analysing the sketch to discover the priority and the nature of each individual data element. High priority elements of the sketch are indicated in figure 1 as bold dashed circles of the middleware.

High priority elements are expected to be requested promptly by the client middleware. However, transmission of data elements is typically managed by *application-specific policies* located in the client middleware, and by using adequate metadata information attached to each individual data element (i.e. priority, access patterns, number of visits, last time read, etc.). Client's requests might also define certain transformations such as compression and transcoding to be applied by the data server on the elements before transmission. Thus, the client middleware can decide not only what elements will be requested, but also in which format and in what order. Once data content is available at the client middleware, it is passed to the application at run-time. Requests for new elements can continue in the background as long as they are needed. The client middleware might even retransmit elements that have been previously shipped with a low quality format (e.g. degraded images) in order to improve data fidelity.

[1]For example, shared libraries in mobile devices represent a problem over low-bandwidth wireless links, precisely because large files (the shared libraries that contain the needed objects) have to be shipped to the mobile device.

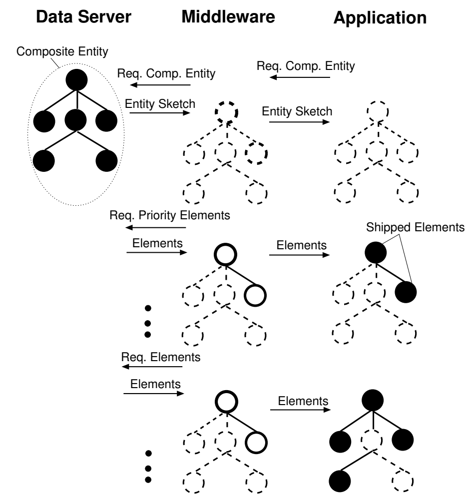

Figure 1: The Process of Shipping Data.

2.3 DATOM – Design Description

2.3.1 Structure of Data

DATOM is designed to deliberately expose the structure of the file. It provides an extensible mechanism to attach relevant metadata to individual elements, such as access patterns and priority. It enables access to portions of a file in order to enforce data distribution at a more granular level. As a consequence, file names in DATOM do not represent a physical abstraction anymore. Instead, they are just the logical entry point to a structured set of data elements.

DATOM does not manage data in the form of monolithic files. Instead, data is stored as *composite entities*. A composite entity is a structured container of *elements*. Elements are the fundamental update unit in the storage system and represent meaningful portions of data for applications. Composite entities are the starting point from which applications can access individual elements or a set of them. The internal structure of the composite element is now dictated by the application and the kind of data that is held. Thus, a document editor might use a tree-oriented representation of its application data, while a composite entity storing bibtex elements or frames for a video player application might use a flatter structure. A sensible degree of granularity imposed on data is an application-specific concern.

Structure of data in DATOM is indicated with XML tags. This provides it with a simple mechanism to indicate structure and to associate metadata to individual data elements by using XML attributes. Structure and metadata represent more powerful hints to enable flexible and efficient adaptation. In fact, the XML structure makes the dynamic manipulation of data much simpler. Addressing and searching of particular branches or elements that need to be communicated between the data server, the client middleware, and applications are performed using the XPath language [8]. Although, there are advantages using a native XML store [9, 10], we decided not to use these tools but rather build our own consistency model specially designed for a mobile world.

2.3.2 The Data Server

Figure 2 shows the architecture of the system. It depicts a storage system specially designed for a heterogeneous environment. The main functions of the data server are update operations on data, access control on a composite entity and their elements, modification of data presentation using transformation services, and to persistently store information. The *Data Storage Manager* is primarily responsible for coordinating them.

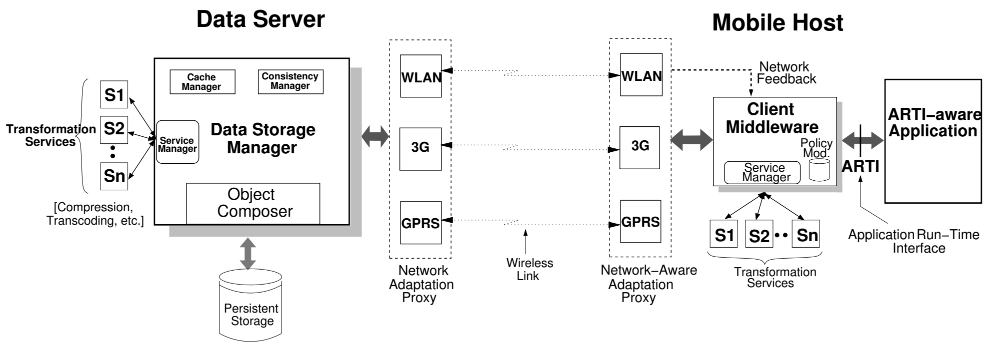

Figure 2: Extended Structured Data Storage System for Heterogeneous Environments (DATOM).

- **Object Composer**. The *Object Composer* stores and retrieves data from persistent storage and organises it to be appropriately manipulated by the *Data Storage Manager*. Even though at the persistent storage layer data is completely passive and static, the *Object Composer* creates a high level composite entity where data elements can be manipulated independently. There are certain disadvantages in using current persistent storage infrastructures. Flat files do not match our structured data representation properly and do not fully decouple the composite entity into its data elements. On the other hand, DBMSs impose rigid schema validations that are not appropiate for the sort of semi-structured data we are interested in. Instead, we considered implementing a low-level storage abstraction following similar ideas to [11], where the structure of data is explicitly used to persistently store it. The benefits of using this approach are: efficient use of disk space, better system performance, and a compatible low-level storage representation.
- **Service Manager**. This part of the system applies transformation services on data. It supports common transformations such as compression (e.g. bzip2) or transcoding of data (e.g. jpeg degradation). We include such support services in our data storage as a repository for generic transformation services. Additionally, other kinds of more specialised and application-specific transformations can be requested if applications register them with the *data server* and the *client middleware*. Transformation services are run as non-blocking processes to improve the performance and throughput of the system.
- **Cache Manager**. The *Data Storage Manager* includes a *cache manager* to improve the performance of the storage system. The cache not only stores elements in their natural representation, but also those elements that have undergone a transformation process. Elements are evicted from the cache using a LRU policy.
- **Consistency Manager**. To enforce consistency the system uses a single-writer multiple-readers model on the whole composite entity. We examined to grant leases [12] on portions of the entity to different clients. This is by far a more efficient and sophisticated method to grant access on data. A leased-based consistency scheme would enable our platform to support interesting data-sharing applications. We consider this sort of more specialised behaviour part of future work. At this stage we want to stress the benefits in terms of adaptation.

2.3.3 Client Middleware

The *client middleware* manages data transmission and adaptation on behalf of the application. Using the sketch received from the storage system, the *client middleware* is capable of requesting the content of individual parts and also track the transmission process of the composite entity. The *client middleware* does not retain any data content of the individual elements.

Using application-specific policies and network feedback available from the *network-aware adaptation proxy*, the *client middleware* decides the order and moment in which elements should be requested, as well as the transformation services that have to be applied to them. In some cases, the transformation process applied at the data server has to be undone within the mobile host (e.g. decompress); the *Service Manager* in the *client middleware* does this job. The knowledge of how specific data types should be handled according to current and future network conditions is kept in the *policy module*.

2.3.4 Network Adaptation Proxy

An *adaptation proxy* is in charge of monitoring the networking context and providing network-feedback to the client middleware in order to enrich the adaptation process and make proactive decisions. It is highly optimised to transport data over a particular network (e.g. GPRSWeb [13]). In order to avoid false triggers from the network, which might cause unnecessary elements requests affecting network performance, the *network adaptation proxy* sends only relevant events to the client middleware using *event filtering techniques*. For example, the network proxy will send network-feedback only when it detects a hard variation in the underlying network conditions such as a drastic drop in the signal strength (averaged over a given interval), a change of network (e.g. WLAN to GPRS), a loss of coverage, or even congestion in the current network.

2.3.5 ARTI-Aware Applications

The sort of interactions between the *client middleware* and the *data server* imposes certain requirements on applications. Firstly, they should be aware of the structure and intra-data relationships of the information they use. This is not an important limitation since this knowledge is already integrated in application's specific libraries. They only need to explicitly indicate a sensible degree of structure that can finally enable the storage system to perform transmission optimisations in favour of them.

Secondly, applications have to be able to receive elements from the *client middleware* at run-time. For this, DATOM makes use of an application run-time interface (ARTI) which represents a specialised tool to enable transparent run-time interchange of information between the client middleware and applications. Many applications can work with partial or degraded copies of data. Thus, ARTI provides applications with the proper mechanism by which user experience can be significantly improved.

2.4 Fine-grained adaptation policies

Networking context is the predominant operational component in heterogeneous environments. The *client middleware* uses its policy module to manage data transmission and to improve performance under hostile conditions. There are two types of policies: *Application-Specific* (ASP) and *System-Supported* (SSP) policies. They are based on meta data and network conditions, respectively.

Imagine a scenario where a mobile host is within a hotspot (i.e. IEEE 802.11b), it solicits a large document rich in content (e.g. figures, text, font styles, etc.) to the data server. DATOM's data server sends the skeleton and grants control of the transmision to the client. Subsequently, DATOM's client-middleware will send data requests considering three elements: mobile host's networking context, policies (i.e. ASP and SSP), and the structure of data represented in the sketch.

Consider the next sequence of events. The **mobile host moves within the hotspot** – Networking context is good; text content is the first element referenced by the sketch. The policy indicates: *"Client CAN request ANY element"*. Consequently, text content is requested from the data server.

Then, the **mobile host enters a WLAN-GPRS transition area** – Networking context is unstable and poor; figures are the next highest priority elements referenced by the sketch. The policy indicates: *"Client MUST request HIGHEST priority elements. IF the element to request is FIGURE, THEN apply a degradation factor of 25%"*. Consequently, figures will be proactively requested from the data server.

3. EXPERIMENTAL TEST SET-UP

3.1 GPRS-LAN-WLAN Testbed

As part of the Cambridge Open Mobile Systems (COMS) project [14], we have implemented a tightly-integrated LAN-WLAN-GPRS testbed (figure 3). It creates a heterogeneous environment for a mobile user serviced by DATOM.

The cellular GPRS network infrastructure currently in use with the testbed is a Vodafone UK's production GPRS network. The WLAN access points (APs) are IEEE 802.11b APs located at different locations of the William Gates Building housing both Computer Laboratory and Laboratory for Communication Engineering of Cambridge University.

The APs provide full in-building coverage to WLAN users. This setup gives us the opportunity to evaluate indoors spaces with nomadic users moving at low speeds. Additionally, access points with extended range antennas have been provisioned to give an outdoor coverage for spatially co-located zones (e.g. an average extended coverage of 500 meters).

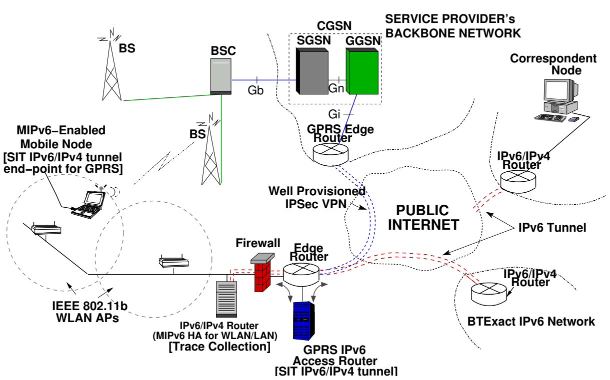

Figure 3: Mobile-IPv6 based LAN-WLAN-GPRS Testbed Implementation.

Furthermore, the GPRS infrastructure offers base stations (BSs) that are directly linked to the SGSN (Serving GPRS Support Node) which is then connected to a GGSN (Gateway GPRS Support Node). A well provisioned virtual private network (VPN) connects the Lab network to that of the Vodafone's backbone via an IPSec tunnel over the public Internet. A separate "operator-type" RADIUS server is provisioned to authenticate GPRS mobile users/terminals and also assign IP addresses. Further information about the testbed is available in [15].

3.2 Experiments over the Testbed

We used the testbed to validate the motivation behind DATOM. For this, we decomposed an OpenOffice document into its elements. We emulated a data repository consisting of decomposed parts of the document such as skeleton, text content, figures, and font style definitions. We assumed these objects have decreasing priorities. Furthermore, we kept degraded copies of elements such as figures to 75% quality degradation. We control data requests from a mobile host using a multimode mobile device having GPRS and IEEE 802.11b connectivity.

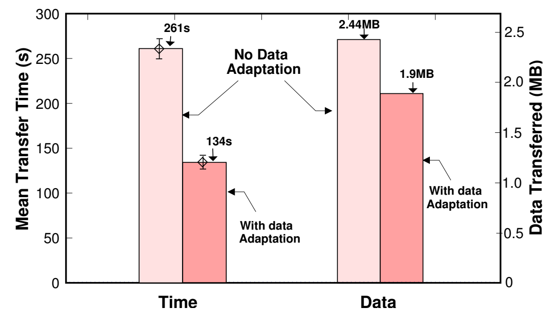

Figure 4: Data Adaptation in a Heterogeneous Environment.

We assume two scenarios: one with data adaptation and another without it. We are able to control requests of individual elements from the mobile client. For this, we used scripts to automate the download process of individual elements and to conduct handoffs (WLAN↔GPRS) at specific time instants during the download. After every WLAN→GPRS handoff, the script used the degraded versions of the images rather than full quality elements. We measured the mean transfer time and the amount of data transferred from over five successful runs for both scenarios. As evident in figure 4, we can find that there is substantial reduction in mean transfer time when we apply data adaptation. This also shows the benefits of fine-grained adaptation in heterogeneous environments, where changes in network connectivity affect data transmission.

4. RESEARCH DIRECTIONS

Research is currently focussed on the implementation of a prototype and its practical validation on our testbed. With the support of transducers, we plan to create an entity-based bank of data for different file formats such as HTML pages, email files, StarOffice documents and PDF files.

We identify *consistency management* and *novel applications* as two fertile areas of future research. We have stressed the need for data adaptation. However, maintenance of consistency under this data model, where fine-grained data sharing can be seen as a feasible alternative, is a significant challenge. Adequate concurrency control have to be analysed [16]. In addition, we are interested in creating applications to expose this features such as a collaborative editing tool for mobile environments.

We are currently working on the definition of policies and algorithms that conform DATOM's data fetching decision model. It aims to provide ubiquitous networking for future mobile users [15]. Moreover, we are exploring further issues related to the integration of heterogeneous environments such as handoff effects.

5. RELATED WORK

Coda [7] file system gives insight into data management for mobile users, typically applicable for environments characterised by disconnected operation. In contrast, we consider a data repository for fine-grained adaptation that exploits the advantages of a different data model. Our data repository is aimed for use in highly heterogeneous environments that ensure an "always-best-connected" operation. Other attempts have been made to try to modify the behaviour of existing file systems initially not designed for mobile environments [1, 2].

Systems such as Odyssey [4] and Rover [17] have examined the benefits of application-aware adaptation. Odyssey introduces type-awareness via specialized components called *wardens*. In our model, network-aware proxies retain application's knowledge in the form of policies and decide on behalf of the application the best way in which data should be shipped.

Techniques such as *subsetting* and *transcoding* have been previously used in different systems [3, 4, 5] in order to adapt the presentation of data and application's behaviour on mobile devices [18]. We attempt to integrate these techniques into the data model.

Puppeteer [5] exploits component-based [19, 20] adaptation in mobile environments. Its strategy is to use applications' APIs to implement application-specific adaptation policies in order to reduce bandwidth usage. Similarly, we argue that knowing the structure of data can be advantageous. Our solution represents an open data model where data logical structure is naturally exposed in the storage system. Furthermore, we are interested in exploring a broader range of network conditions in order to provide a comprehensive data adaptation model.

An important amount of work has been done in systems such as Bayou [21], GLOMAR [22], and Coda [7] to maintain consistency in a mobile storage system. Our consistency model will consider results from these systems.

In [23], user profiles are used to express information requirements. Using this profile, it is possible to specify the data the user is interested in and its relative worth. This approach is similar to ours in the sense that we also consider filtering data as an important factor while shipping data to a mobile device.

Network-aware adaptation proxies such as GPRSWeb [13] offer a complete solution for an efficient transport over GPRS. It uses a new transport protocol, called GPRSWeb, which is highly optimised for use over GPRS. In the same way, network-aware adaptation proxies can also be designed for WLANs and 3G if required.

6. CONCLUSION

In this paper, we have proposed DATOM – a structured data storage system for mobile data management. DATOM attempts to provide applications with a generic method by which they can explicitly indicate adaptation strategies. This system significantly departs from the concept of files as objects to a fine-grained data abstraction. This abstraction is then used as an invaluable tool to dynamically adapt in the presence of changes as posed by heterogeneous mobile environments. Our ongoing research aims to implement and thoroughly evaluate the benefits of DATOM over our WLAN-GPRS-LAN testbed.

Acknowledgements

We wish to thank Ian Pratt for providing the support and necessary advice for this project. This research has been supported by Vodafone Group R&D, Sun Microsystems Inc., and Benchmark Capital. Calicrates Policroniades and Pablo Vidales have scholarships from the Mexican government through the National Council of Science and Technology (CONACYT).

7. REFERENCES

[1] F. Andre and M. Segarra. On building a file system for mobile environments using generic services. In *12th International Conference on Parallel and Distributed Computing Systems*, Radisson Bahia Mar Beach Resort, Florida, USA, August 1999.

[2] Cynthia D. Rais and Satish K. Tripathi. Measuring NFS performance in wireless networks. Technical Report CS-TR-3582, University of Maryland, 1995.

[3] A. Fox, S. Gribble, Y. Chawathe, and E. Brewer. Adapting to network and client variation using active proxies: Lessons and perspectives. In IEEE Personal Communications on Adapation special issue, 1998.

[4] Brian D. Noble, M. Satyanarayanan, Dushyanth Narayanan, James Eric Tilton, Jason Flinn, and Kevin R. Walker. Agile application-aware adaptation for mobility. In *16th ACM Symposium on Operating Systems Principles (SOSP)*, pages 276–287, Saint Malo, France, 1997.

[5] Eyal de Lara, Dan S. Wallach, and Willy Zwaenepoel. Puppeteer: Component-based adaptation for mobile computing. In *3rd USENIX Symposium on Internet Technologies and Systems*, pages 159–170, March 2001.

[6] Nigel Davies, S. Pink, and Gordon S. Blair. Services to support distributed applications in a mobile environment. In *1st International Workshop on Services in Distributed and Networked Environments*, pages 84–89, Prague, Czech Republic, June 1994.

[7] M. Satyanarayanan, James J. Kistler, Puneet Kumar, Maria E. Okasaki, Ellen H. Siegel, and David C. Steere. Coda: A highly available file system for a distributed workstation environment. *In IEEE Transactions on Computers*, 39(4):447–459, 1990.

[8] James Clark and Steven DeRose. XML path language (XPath). Technical report, W3C, November 1999.

[9] Tamino xml server. `http://www.softwareag.com/tamino`.

[10] Xindice. `http://xml.apache.org/xindice/`.

[11] Carl-Christian Kanne and Guido Moerkotte. Efficient storage of XML data. In *Proceedings of the International Conference on Data Engineering (ICDE)*, 2000.

[12] Cary G. Gray and David R. Cheriton. Leases: An efficient fault-tolerant mechanism for distributed file cache consistency. In *Proceedings of the Twelfth ACM Symposium on Operating Systems Principles*, pages 202–210, 1989.

[13] Rajiv Chakravorty, Andrew Clark, and Ian Pratt. GPRSWeb: Optimizing the web for GPRS links. In *Proceedings of the ACM/USENIX First International Conference on Mobile Systems, Applications and Services (ACM/USENIX MOBISYS)*, San Francisco, USA, `http://www.cl.cam.ac.uk/users/rc277/gprsweb.ps`.

[14] Cambridge Open Mobile Systems (COMS), Project Initiative. `http://www.cl.cam.ac.uk/Research/SRG/netos/coms/`.

[15] Pablo Vidales, Leo Patanapongpibul, and Rajiv Chakravorty. Ubiquitous networking in heterogeneous environments. In *8th IEEE Mobile Multimedia Communications (MoMuC), (to appear)*, Germany, 2003. `http://www.cl.cam.ac.uk/users/rc277/ubiq.pdf`.

[16] Ashok M. Joshi. Adaptive locking strategies in a multi-node data sharing environment. In *Proceedings of the 17th International Conference on Very Large Databases*, pages 181–191, Barcelona, Spain, September 1991.

[17] Anthony D. Joseph, Joshua A. Tauber, and M. Frans Kaashoek. Mobile computing with the rover toolkit. *IEEE Transactions on Computers*, 46(3):337–352, 1997.

[18] F. Kaashoek, T. Pinckney, and J. Tauber. Dynamic documents: Mobile wireless access to the WWW. In *IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, Santa Cruz, California, USA, 1994.

[19] Structured Data Storage. `http://msdn.microsoft.com/library/`.

[20] Eyal de Lara, Dan S. Wallach, and Willy Zwaenepoel. Opportunities for bandwidth adaptation in microsoft office documents. In *Proceedings of the 4th USENIX Windows Symposium*, Seattle, Washington, USA, August 2000.

[21] A. J. Demers, K. Petersen, M. J. Spreitzer, D. B. Terry, M. M. Theimer, and B. B. Welch. The Bayou architecture: Support for data sharing among mobile users. In *IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, Santa Cruz, California, USA, 1994.

[22] Simon Cuce and Arkady Zaslavsky. Adaptable consistency control mechanism for a mobility enabled file system. In *Proceedings of the Third International Conference on Mobile Data Management (MDM'02)*, 2002.

[23] M. Cherniack, M. Franklin, and S. Zdonik. Expressing user profiles for data recharging. *In IEEE Personal Communications: Special Issue on Pervasive Computing*, August 2001.