

An Architecture for Sharing, Discovering and Accessing Mobile Data and Services: Location and Mobility Issues*

Christopher Ververidis, Stratis Valavanis, Michalis Vazirgiannis, George C. Polyzos

Mobile Multimedia Laboratory
Department of Informatics
Athens University of Economics and Business
Athens 104 34, Greece

{chris, valavani, mvazirg, polyzos}@aueb.gr
<http://mm.aueb.gr/>

Introduction

There is an increasing number of autonomous, portable, computing devices that we use in everyday life, work and leisure alike, such as notebook computers, personal digital assistants (PDAs), palmtops, mobile phones, digital cameras, etc., including combination devices. As a result, a great amount of “mobile” data resides on these devices in a distributed, non-integrated manner. Similarly, services not only can be made available *to* mobile devices, but *by* mobile devices.

These resources are not only heterogeneous but also carry context-dependent semantics, depending on user, location, community and other parameters. Furthermore, they exhibit variable availability over time, depending on mobility parameters and network coverage. Alternatively, they can always be accessed locally, by using the device directly. As a result, a virtual super-database with a dynamic schema is formed and many new issues concerning its use arise. Traditional database (DB) management, querying, caching and indexing techniques are not directly applicable in this context and new paradigms are needed.

Many research projects deal with (and have addressed in the past) issues in distributed computing environments and the problems of sharing, locating and accessing resources from files and storage [6,9,2] to CPU cycles [3] and network connectivity. We are involved in such a project, DBGlobe, which we describe briefly in the next section. The DBGlobe project takes a data-centric approach, i.e., it attempts to find innovative ways to dynamically describe, integrate, index, cache, search, retrieve data and services in such an environment in a uniform and optimal way, transparently to the user, in order to offer increased availability of the resources, scalability and fault-tolerance.

Users carrying mobile devices want to share data and services. These resources are diverse and do not conform to specific rules regarding types of data, data representation, etc., making it practically impossible for other users to access them effectively to obtain information and services.

The main aim is to put an order in this chaotic pool of information. This is achieved by making data accessible via services. This offers a uniform, universal representation of the way to access the data that is understandable by other peers, can be stored, processed, compared and classified in taxonomies and is possibly reusable and customizable. The use of service ontologies facilitates searching for partially matching requests or resolving concept conflicts and locating term similarities or relationships.

* This research is supported by EU IST project DBGlobe (IST-2001-32645) under the FET Proactive Initiative on Global Computing for the co-operation of autonomous and mobile entities in dynamic environments.

Another issue is that the set of services that can be used to fulfil a user's request is enormous and it might be difficult for the user to specify exactly what he is looking for. This ambiguity can be reduced by employing context awareness. One of the parameters related to context is *location*. By exploiting location information about the data or service provider and the data or service requestor, irrelevant services and information can be quickly filtered out before reaching the requestor, saving time and bandwidth and leveraging the overall quality of service.

The DBGlobe Project

DBGlobe is a European Information Society Technologies (IST) project under the Future and Emerging Technologies (FET) initiative for the co-operation of autonomous and mobile entities in dynamic environments. The project is a joint effort of the *University of Ioannina*, the *Institut National de Recherche en Informatique et en Automatique* (INRIA), the *Computer Technology Institute*, the *Athens University of Economics and Business*, the *University of Cyprus*, the *University of California Riverside*, the *Technical University of Crete* and *Aalborg University*.

The project aims at developing novel data and metadata management techniques to deal with the challenge of global computing with a **data-centric** approach. The traditional database approach of collecting, caching and indexing data of interest in monolithic database management systems becomes obsolete in modern global computing environments. Source data are no longer homogeneous, they exhibit high degrees of distribution, the network topology is dynamic, the communications may exhibit unreliable and unpredictable quality of service and the context of data is of more importance to the end-user in order to locate the "target" information. This creates the need for new theoretical foundations in all aspects of data management: modeling, storage and querying.

DBGlobe:

- considers mobile entities as primary data stores as well as mini-servers, i.e., computational entities that protect and encapsulate access to their data, and
- broadens the database data management focus to address the issues of mobility, autonomy, incomplete information, scale and adaptation, which arise in dynamic environments.

Proposed Architecture

The proposed infrastructure architecture is designed considering the following observations:

- A great number of user queries (in certain application domains) are of local interest in the sense that the user needs fast responses (and not necessarily complete), based on nearby resources, to fulfill immediate needs.
- Much information and many service requests can be served by peers, which wish to manage their shared resources independently of some central authority. Nevertheless, they do expect them to be widely published and accessible.
- Pure P2P architectures, where all the functionality is embedded in the peers and no central administration exists [4], impose high overheads to querying and discovery. Demanding a query to perform many hops greatly increases the number of required messages, flooding the network.
- Hierarchical architectures are an alternative solution if we are interested in requests that need global resources to be served. However, they demand a large layered infrastructure and they might cause a significant delay in answering simple requests.

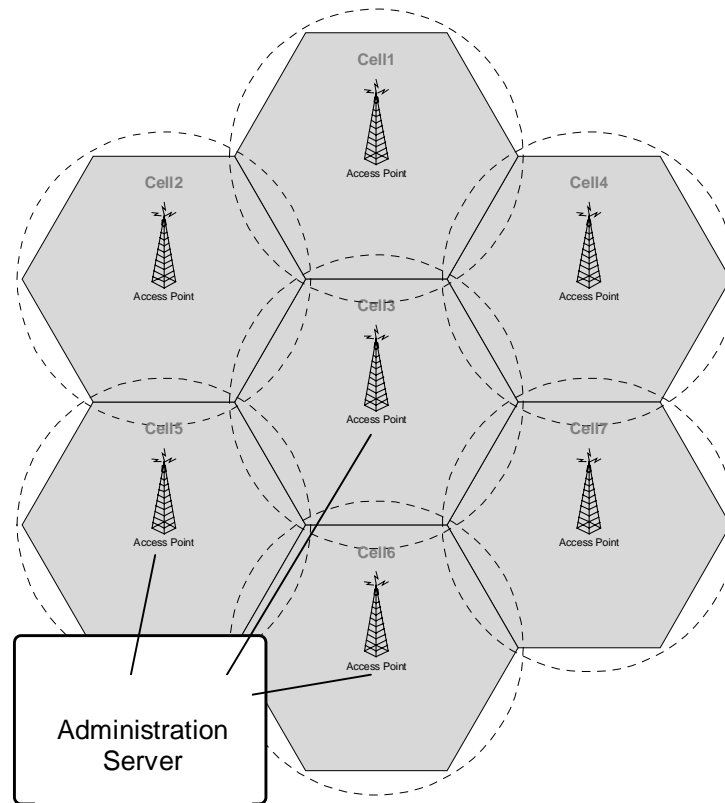


Figure 2: Cell based space model

Key Terminology

Primary Mobile Object (PMO) is a basic, mobile, autonomous, computing device capable of communicating independently with the administration server of a cell, via some communication medium (Internet, LAN, VPN, and in general any wired or wireless network). A PMO is the elementary environmental entity that the DBGlobe framework interacts with.

A **cell** is a geographical area served by an Administration Server. The layout of cells is assumed predefined and static.

An **Administration Server (AS)** is an autonomous system that manages the **cell** and the system aspects of the PMOs that are in the cell at any given moment. Each cell is populated by large numbers of PMOs acting as **resources** or **requestors** of service. AS functionality includes network addressing, managing accessibility, requests handling, service publishing, service description caching, etc.

A service is accessible through messages, language and platform independent, and produces output messages (results) which are machine-oriented. We assume that a service is:

- described using a standard description language (e.g., WSDL or DAML-S [12])
- published or advertised through a widely acceptable mechanism (e.g., UDDI directory or ebXML registry [13])
- discovered through a widely acceptable mechanism.
- invoked over the network through its declared API
- possibly composed by using other services

Services can be used to encapsulate data and provide a customizable way to access them.

The Service Ontology we use is a global, fixed, terminological classification of services. The service categories are not necessarily fully described using concept definitions or axioms. Instead, they are partially specified using subtype-supertype or part-whole relations, which determine the relative positions of the concepts with respect to one another (e.g., similar to WordNet [11]).

We propose a hybrid (partially ad-hoc) architecture, where geographical 2-D space is divided into adjacent administrative areas (similar to GSM cells), each managed by an Administration Server. Each server can independently manage the PMOs which enter its own area of authority and is capable of performing login management, local network address management, session management, local service registration, semantic matching of requests to service descriptions, submission of request subscriptions, etc. The data flows between the cell ASs include: (a) extension of queries in neighboring ASs (possibly upon explicit user request), (b) forwarding the list of neighbors, (c) PMO tracking information, (d) pushing service description files (proactively and on demand), and (e) pushing subscription definitions (proactively and on-demand).

It is obvious that there is no horizon for user requests within the boundaries of the users' current cell, and in cases where this proves to be insufficient the user can extend his request to adjacent cells at reasonable response time cost, thus covering a large area around his location. The disadvantage is that a request looking for services globally would need to go on extending the query to a larger "ring" of cells around the current location. However, under the assumption that queries are of local interest the requestor will soon receive a response regarding the desired services within his cell and in a reasonable time a response regarding the services in adjacent cells. A similar process would be significantly slower in the case where the ASs were not independent and would have to forward the request to a hierarchy of service indices.

Mobility Issues

In distributed architectures, like the one described in this paper, many issues related to mobility arise. The first issue has to do with frequent disconnections and highly variable bandwidth. Today's computer systems depend heavily on the network and may cease to function during network failures if not carefully designed. Network disconnections (i.e., network service failures) are more frequent in mobile computing than in fixed wireline systems, because wireless communication is so susceptible to failures [5]. In fact disconnection can be considered as an extreme case of the network bandwidth variability, which is continuously observed in wireless systems. However, adaptation techniques can address the problem of bandwidth variability to some extent, particularly for real-time multimedia communication. In the case of disconnections, caching can provide an effective solution. However, caching raises cache consistency as a prominent issue. Also, in the context of services, caching gets a totally different meaning. In this context the issue arises whether services could be integrated to the mobile unit in order to allow stand-alone operation, or be replicated to some degree locally. Therefore, the challenge is how to exploit all the above-mentioned techniques in this new context.

Our view of disconnections is the following: we consider that disconnection occurs due to problems in the physical layer of wireless communication or due to the movement of the mobile units between cells. For example let us assume that a mobile unit offering a specific service in a given area moves out from it. Then all the other units that used that service can consider that they were disconnected from it. Automatically the system tries to find a similar service offered in the area of the "disconnected" mobile units in order to "rebind" them to it. Caching is also used for service descriptions as described in the previous section. So even when a mobile unit gets disconnected, the descriptions of the services that it offers are maintained in its AS and can be viewed by other mobile units accompanied with predictions of when these services will likely reappear (predictions are derived using a mechanism that exploits mobility patterns).

A second issue is the principle of locality. According to this principle mobile units should be served preferably by mobile units located near to them. In mobile systems the devices are relatively weaker than hard-wired devices and bandwidth is usually much lower than in wired networks. In our proposed architecture where we have a global-scale system it would be not efficient or even possible to let any mobile unit to issue requests on any other unit around the globe. The set of answers would be outrageous for such a device to handle and the communication cost too high. In order to overcome this problem we propose a discovery mechanism that facilitates local area search for deriving answers to requests, making the additional assumption that users are generally more interested for services in their proximity (e.g. services located at the same area served by an AS or services in all adjacent areas in the network topology).

A final issue, closely related to the previous one, which arises from the fact that requestors are mobile, is that the answers to a request may differ depending on the location from which the query originated. So if a mobile unit issues a request when located in Area A and before taking the results it moves to Area B, then these answers will be out of date. Our approach is that the request is recomputed by the system based on the new location (Area B). In this way we use context information (location is a type of context) to leverage the quality of the provided answers.

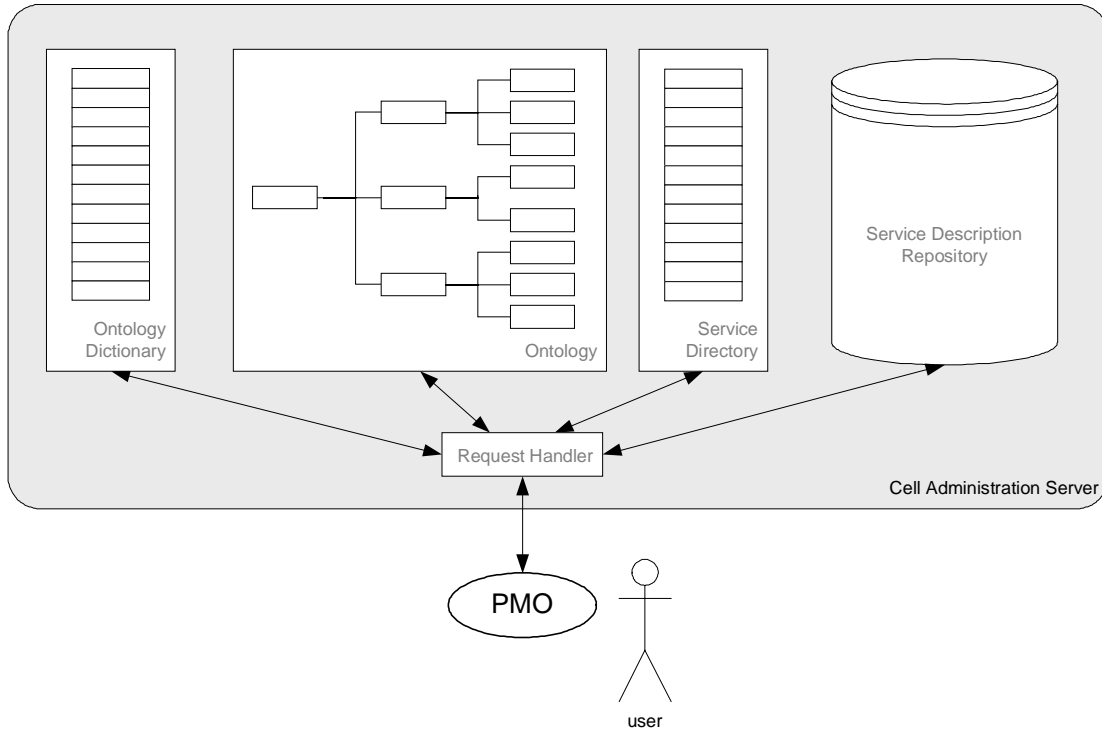


Figure 3: User-Defined Requests

Location Issues

Let us assume that a given mobile user wishes to submit a specific request to the system (e.g., she wants to locate services that allow her to find transportation in her area). The user does not know in advance which services are available at the moment neither the necessary interfaces. Therefore the first step is to understand what the user is looking for and offer her a set of alternative services that provide this kind of output. This would not be possible by simply using WSDL [7] since WSDL would just enable searching for services that include exact words such as “Taxi” and “reservation” in the name and short

description and check whether the service accepts input messages about location and destination and returns reservation confirmation number.

Therefore, we provide the PMO with a specially designed tool, the Service Request Interface, to assist the user in describing the request in a formal manner assisted by the global ontology of services. The set of words provided by the user (w_1, \dots, w_k) are matched with the dictionary of terms (thesaurus) to find exact lexicographical matches. In case of no match at all, the user is asked to issue a new request. The dictionary works as a fast index to the ontology (to avoid searching the whole ontology) for every request.

The matches in the dictionary lead to several points in the ontology where different instances of the words are found. Using a semantic distance measure [10], we select the part of the ontology where the set of words appear closer to each other (one as a higher level category and the others as subcategories or relevant categories). Thus we can expect that we located the right context in which the user meant to use the set of words.

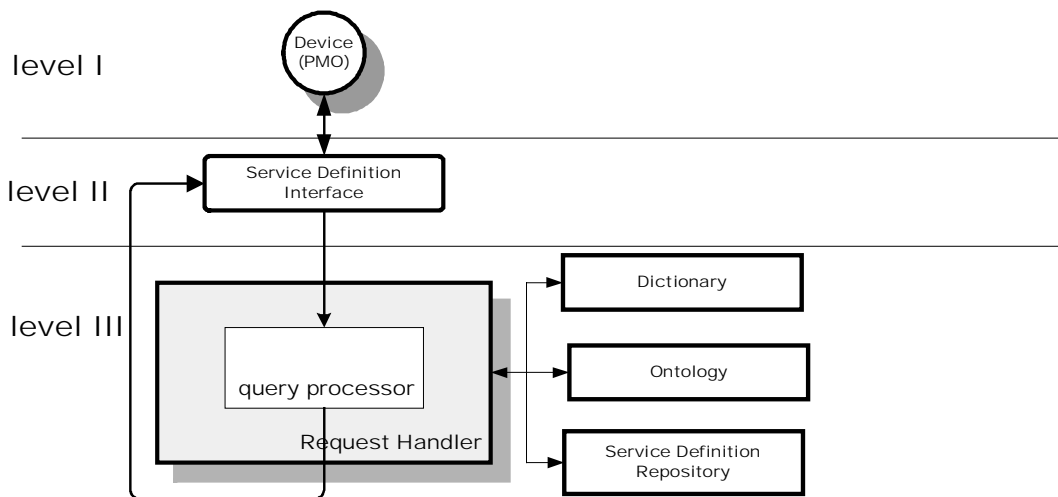


Figure 4: Request Handler

As an example, assume that the user is issuing a request for “travel, book, taxi.” One appearance of these words in the ontology is under “travel->reservation (synonym: bookings)->taxis” and the other under “Books->travel->city-guides->(relevant)->mass transportation information->taxi yellow pages.” The matching mechanism should prefer the first appearance as more relevant. Having chosen one path in the ontology we traverse the categorization down to the leaves and select all the services that are categorized there. Here is where the filtering using location information takes place.

If the system replied to the above query without taking into account location information, then the requestor PMO would be flooded with irrelevant information about all possible taxi services, anywhere in the world. Our design allows for every “candidate” service returned by the search in the global service ontology, for the location of that service to be compared to the location of the requestor PMO. If there is a “proximity match” then the service is returned, otherwise it is rejected.

However, there is a possibility that matched services may not be currently active in the area. For example, a service, which used to be regularly available in an area, could unexpectedly disappear from that area due to network problems or due to moving to another location. The service description is cached in the registry of the AS and thus it could be returned after a successful matching. The set of currently

inactive services can be useful to the requestors, provided that they exhibit a high probability of reappearing soon in that area. Therefore, we maintain a service appearance log in every given administrative area in the form of a multidimensional Object Exchange Model graph and employ a pattern discovery mechanism that constantly calculates the probability and the expected time of re-appearance of inactive services in the area based on their past behavior [8]. Thus, we try to predict the availability of services by exploiting regular habits of users, which affect the availability of PMOs. For example, a user could provide a service only when he is in the office, i.e., Monday to Friday, during working hours. Such a service request, issued at night, would normally get no results, while in our case the requestor would be informed that according to cached information, in that certain location it is highly probable that the requested service will be available the next morning.

Conclusion

We described a hybrid architecture that enables mobile devices to share their data encapsulated in services. This allows access to them by other devices, while at the same time it provides a publishing and advertising mechanism that facilitates semantic resource discovery. An additional location based filtering mechanism for responses protects the requestor PMO from flooding with mostly irrelevant information. Finally, “area-based service behavior profiling” is used as a solution against unexpected disconnections and partially available resources with traceable appearance patterns.

For our future work we are planning to extend the architecture by integrating fallback techniques to support ad-hoc system operation in isolated groups of PMOs in cases where communication with an AS cannot be established. The implementation of a testbed for the proposed architecture using WLAN technologies in order to implement the cell-based space model is already underway.

References

- [1] T. Imielinski and S. Goel, “DataSpace: Querying and Monitoring Deeply Networked Collections in Physical Space,” *IEEE Personal Communications*, 7(5):4-9, October 2000.
- [2] J. Kubiawicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao, “OceanStore: An Architecture for Global-Scale Persistent Storage,” Proceedings of the Ninth international Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2000), November 2000.
- [3] I. Foster, C. Kesselman, S. Tuecke, “The Anatomy of the Grid: Enabling Scalable Virtual Organizations,” *International Journal of Supercomputer Applications*, 15(3), 2001.
- [4] S. Saroiu, P. Krishna Gummadi, S.D. Gribble, “A Measurement Study of Peer-to-Peer File Sharing Systems,” Proceedings of Multimedia Computing and Networking 2002 (MMCN '02).
- [5] G.H. Forman, J. Zahorjan, “The Challenges of Mobile Computing,” *IEEE Computer*, 27(6), April 1994.
- [6] I. Clarke, O. Sandberg, B. Wiley, and T.W. Hong, “Freenet: A Distributed Anonymous Information Storage and Retrieval System in Designing Privacy Enhancing Technologies,” International Workshop on Design Issues in Anonymity and Unobservability, H. Federrath, ed., 2001.
- [7] Web Services Description Language 1.1, W3C Note, <http://www.w3.org/TR/wsdl>, March 2000.
- [8] R. Goldman, S. Chawathe, A. Crespo, J. McHugh, “A standard textual interchange format for the Object Exchange Model (OEM),” Technical Report, Stanford University, October 1996.
- [9] M. Ripeanu, “Peer-to-Peer Architecture Case Study: Gnutella Network,” Proceedings of 1st IEEE International Conference on Peer-to-Peer Computing (P2P 2001), Linkoping Sweden, August 2001.
- [10] A. Rodríguez and M. Egenhofer, “Determining Semantic Similarity Among Entity Classes from Different Ontologies,” *IEEE Transactions on Knowledge and Data Engineering* (to appear).
- [11] G. Miller, R. Beckwith, C. Fellbaum, D Gross, and K. Miller, “Five papers on wordnet,” Technical report, Stanford University, 1993.
- [12] A. Ankolenkar, M. Burstein, J.R. Hobbs, O. Lassila, D.L. Martin, D. McDermott, S.A. McIlraith, S. Narayanan, M. Paolucci, T.R. Payne and K. Sycara, “DAML-S: Web Service Description for the Semantic Web,” Proceedings of the First International Semantic Web Conference (ISWC), 2002.
- [13] The ebXML website, <http://www.ebXML.org>