

System integration for mobile computing and service mobility

N. Diehl, D. Grill, A. Held, R. Kroh, T. Reigber, T. Ziegert
Daimler-Benz AG, Research Center Ulm, Information Technology
Wilhelm-Runge-Straße 11, P.O. Box 2360, 89013 Ulm, Germany
Tel.: +49 731 505 2132
Fax.: + 49 731 505 4218
e-mail: {diehl,grill,held,kroh,reigber}@dbag.ulm.DaimlerBenz.COM

Abstract

In future networks, mobile computers will become increasingly important. Distributed, mobile applications enable information access anywhere, anytime. Mobility, location independence and adequate information access will be supported simultaneously. Major problems due to mobility and the specific characteristics of portable computers and wireless communications are dynamic configuration, moving resources, reachability, data and functional consistency. In this paper we present a system called MOBI-DICK that addresses the problems of mobile computing. The requirements and solutions for service mobility are described in detail.

Keywords

Mobile computing, wireless communications, service trading, mobile information systems

1 INTRODUCTION - DISTRIBUTED, MOBILE APPLICATIONS

Mobile computing systems allow to access and to distribute information at any time and anywhere. New portable computers and wireless communications technologies particularly enable mobile computing. Additionally, many system integration effects have to be addressed to support mobile users and mobile computing applications. There are at least three important issues that determine mobile computing systems:

- A basic but far-reaching fact is that the users are moving and thus the topology of the system is dynamically changing. This includes user- and terminal mobility.

- The mobile terminals will always be less powerful than the stationary computers. Thus we have a performance disparity between the mobile and stationary computers. That has a strong influence on the system design, e.g., workload balance, etc.
- The data rates of wireless connections will always be clearly lower than those of wired connections. Additionally, wireless connections are not as reliable as wired.

Major problems due to mobility and the specific characteristics of portable computers and wireless communications are dynamic configuration, moving resources, reachability, data and function consistency. These problems have to be treated in more than one of the communication layers. A full system architecture is necessary to support mobile users and mobile computing applications (DiHe 94, DGH 94, Dieh 95). Additionally security aspects are very important. User management and work flow management are necessary to support concrete applications and work flows with mobile users.

This paper discusses selected aspects of mobile computing. Section 2 discusses general aspects of system integration. In section 3 we introduce MOBI-DICK (**MOBI**le **Data** **InterChange** **Kit**) a system to support distributed mobile applications. Section 4 gives an example, how service mobility is supported. Finally, in Section 6 our test system is described.

2 SYSTEM ISSUES OF MOBILE COMPUTING

In the following we discuss several new challenges that are important when designing systems with mobile users (see also Weis 93, DiHe 94, DGH 94, ImBa94 and Dieh 95).

2.1 Mobility models

Essential tasks of mobile computing systems are the handling of terminal- and user-mobility as well as the transparent access to information and services. This also includes service mobility. Different kinds of mobility should be regarded:

- Terminal mobility - Due to the mobility of the portable computers their network access points to the fixed network might change. However the user should not be concerned with these problems, especially the addressing concepts. For the TCP/IP-protocol stack Mobile-IP (IPMS 95) deals with these issues.
- User mobility - A user can move with his mobile terminal or logging on different computers (stationary or mobile) in the network and find his accustomed environment.
- Service mobility - If a mobile moves between different networks the available services may change depending on the (sub)net the mobile is currently registered at. Additionally if the mobile provides services to other computers the system topology is changing too. This must be supported by an extended arbitration of services

Different mobility models such as 'lazy professor', 'pop-up' or 'travelling salesman' have to be considered.

2.2 Configuration and resource management

Mobile computers dynamically move between subnetworks and organizational domains. Additionally they may be connected to different networks such as fixed networks, radio LANs or radio WANs. Therefore, the configuration and topology of the system are dynamically changing. This affects the reachability of users as well as the reachability of available services.

Systems that support mobile distributed applications aim at providing transparency of those dynamic features. Hence mobile and fixed applications should have the same functionality and performance characteristics. The user should not have to cope with the specifics of different locations.

Transparency can be supported by additional functionality in different layers of the communication system. By extending the IP-protocol stack, Mobile-IP (IPMS 95) allows transparent migration of nodes within the internet. This location transparency however, does not solve all the problems of mobile computing.

The need for the same user-environment, independent of the current location, i.e., the mapping of locally available resources and services to the ones used in the user's home location is not supported by Mobile-IP. Profile management and dynamic reconfiguration of work-environments, the need for concurrency and consistency and the administration of access rights require new concepts and solutions.

2.3 Wireless communication systems and quality of service

As mentioned before the mobile computers can be connected over different networks such as fixed networks, radio LANs or radio WANs (GSM or Modacom/ARDIS or Mobitex). The quality of service (QoS) attributes such as throughput, error rate, latency, communications costs, connection oriented vs. connection less, ... differ significantly. Mobile computing systems should incorporate these aspects and adapt to and support different wireless and wired communication systems. The application level should provide an established and widely accepted communication mechanism and API such as RPC or socket interface. The QoS parameters should be configurable depending on the concrete applications.

2.4 Data distribution and autonomous operations

The migration of portables and the lower performance of mobile computers and wireless communications systems lead to new requirements for information- and data management in distributed systems. New adaptable algorithms are needed to handle the distribution of communication and processing load. To meet the required quality of service, applications must adapt dynamically to the characteristics of the available communication system. Thus the workload should not be shared equally between mobile and fixed computers. New workload sharing strategies are needed. To support both data-intensive and process-intensive applications and services, special strategies for caching (client vs. server caching) and special tools or services for reconnection and recovery are needed.

Another typical feature of mobile computing applications is frequent disconnection of the radio link. Sometimes it could be even advantageous, not to transmit data to save battery

power. For both cases (planned and unplanned disconnection) new transaction - and caching algorithms had to be developed (Saty 93).

2.5 Security aspects

The provision of IT-security is very important in mobile computing systems. Besides problems of eavesdropping and disclosure of sensitive information stored on the mobile stations user authentication and access control are crucial issues. Based on a distributed authentication scheme that introduces different levels of trust, security services for access control, auditing, etc. have to be implemented as part of the operating system.

3 MOBI-DICK

The project MOBI-DICK (DiHe 94, DGH 94) of Daimler Benz Research Ulm aims to develop a system model and architecture that meets the requirements mentioned above. A pilot system is implemented to demonstrate the feasibility of the concepts. A similar approach is used by TU Dresden (ScKü 95).

MOBI-DICK is a platform to develop distributed applications with mobile components in heterogeneous systems. It provides:

- A model for mobile computing systems with an application oriented view that helps to design future mobile applications
- A high level shell or application program interface (API) where the special needs and requirements of wireless communication are transparent to the application programmer. MOBI-DICK supports program development for distributed applications with mobile and fixed hosts.

The major goals of MOBI-DICK are:

- terminal mobility - user mobility - service mobility
- location dependent services - location independent services
- ad-hoc networks
- autonomous work during disconnection
- IT-security (authentication, authorization, ...)
- adaptation and support of different wireless and wired communication systems

Our mobility model is a pop-up model, that means there is no a priori information at which place and at which time a user will use system resources. However if there is any additional knowledge, this should be used explicitly.

3.1 System architecture of MOBI-DICK

The MOBI-DICK Shell introduces a new layer in the sense of ISO/OSI Reference Model on top of the OSI transport layer. Figure 1 shows its internal structure.

The system model is depicted in Figure 2. The dashed lines indicate how this can be mapped into hardware (WAN, LAN, etc.). The computers (both wire-based and wireless connected)

are modeled as *stations*. Stations are the basic components of MOBI-DICK: they provide the mobility management for the applications. There are no dedicated servers and clients. Each station acts as a server (i.e., provides services for other stations) and as a client (i.e., uses services of other stations). The stations are grouped in several disjoint administrative areas, so-called *domains*. A domain is an organizational unit for administrative purposes. A domain can be looked upon as a logical cell. Thus domains are similar to DECE cells. All domains together represent the system's infrastructure. By adjusting the size and the number of the domains, the system scales for scenarios with a large number of mobile stations. The location information of MOBI-DICK is bound to the domains. This means: the location of a station is either within a domain or the station is disconnected.

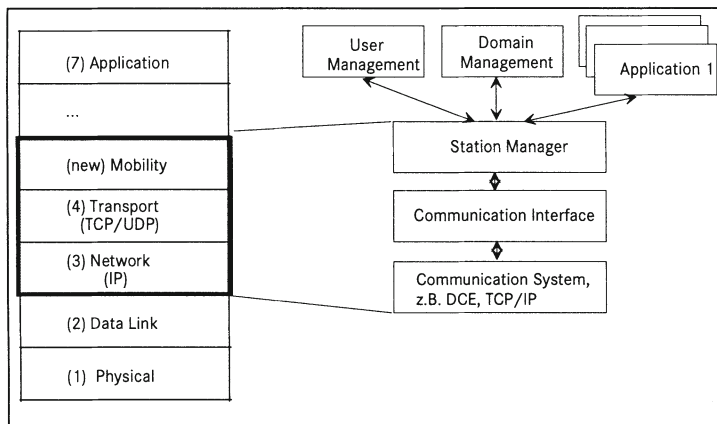


Figure 1 Software architecture of MOBI-DICK.

Each domain is managed by a *Domain Manager* (DM). The domain manager holds and manages information about the state of the domain and its stations. The DM knows the location of the stations (especially of mobile stations), the attributes of the stations, and the mapping of users to the stations. It holds a list of all resources available within the domain and their status. For example it knows which stations are switched on and reachable, which are in doze mode, and what the attributes of those stations are. Moreover it supports the active applications within the domain and the communication means to reach other domains.

3.2 System components

Station Manager

The main component is the *Station Manager* (see Figure 3). This is an active component that represents MOBI-DICK on this station. The station manager (SM) provides the interface for the applications and knows the station's attributes and the available communication systems. This information is stored in the *Station Manager MIB* (Management Information Base) (SM-MIB). The station manager provides:

- Security Management for user administration. This includes authentication and authorization between user and station as well as between station and domain.
- Resource Management to handle the stations resources such as CPU-power, storage capabilities, etc. and the services that are provided by this station.
- Domain Management to handle the communication and the protocols with the domain manager (e.g., registration/deregistration in a domain, import/export of services).
- Communication and Disconnection Management to handle communication between stations and deal with the problems of disconnection.
- Additional services to support local applications are print- and file-transfer services.

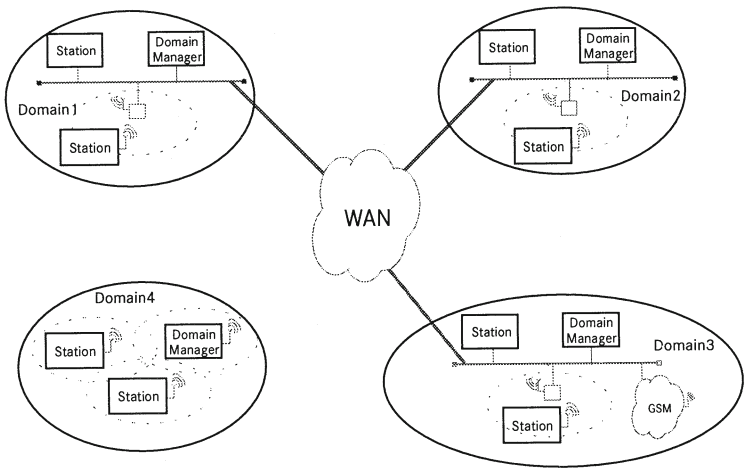


Figure 2 MOBI-DICK system model.

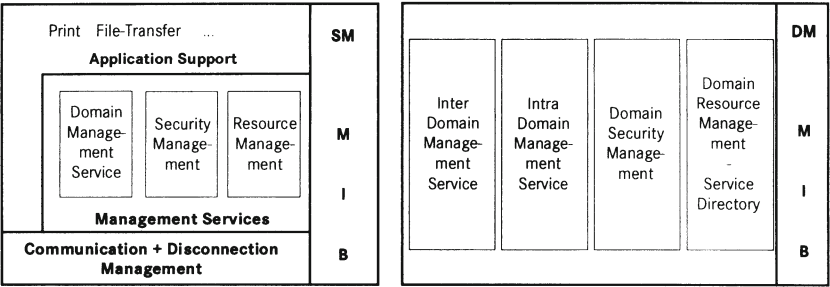


Figure 3 Structure of the station manager and the domain manager.

Communication Interface

The *Communication Interface* (CI) maps MOBI-DICK messages to the underlying communication system (e.g., TCP/IP, OSF/DCE (OSF 92), Mobile-IP). If there are special demands indicated by the kind of communication system (e.g., channel allocation in some radio

networks), the communication interface takes care of them. If there is more than one possibility to transmit messages, the CI can choose the appropriate one according to the information from the SM. The communication protocols are based on a MOBI-DICK specific message format.

Domain Manager

The structure of the domain manager resembles the structure of the station manager. His tasks however are slightly different. The domain manager is responsible for the whole domain. He knows about available resources and services, network topology and configuration as well as properties of the communication links. The necessary information is stored in the *Domain Manager MIB* (Management Information Base) (DM-MIB). The domain manager provides the following services:

- Security Management for user administration within the domain, authentication and authorization of stations to domain manager and between domain managers. Security Services within a domain are provided by the domain manager and the involved station managers. Security services between domains are provided by the domain managers. A set of basic security services is implemented as an integral part of MOBI-DICK and is available for all applications.
- Resource Management for administration of the domain's resources and services. For the latter, a so called Service Directory (SD) is used. Here information about service-class, type of service, location and access rights are stored.
- Intra-Domain Management handles the communication to stations in this domain, i.e., Registration/Deregistration, etc.
- Inter-Domain Management covers the communication between domains.

In principle, any station can act as domain manager. Generally the domain manager is a powerful fixed host.

Application Manager

The task of the *Application Manager* (AM) is the administration of MOBI-DICK applications. These can be 'simple' applications such as database queries, creation/printout of document as well as 'complex' distributed applications such as joint editing in ad-hoc work groups. There is one application manager for every application. The application manager knows what services and resources are needed for his application and is responsible that they are available. Additionally the application manager maintains the consistency of the application data and process states by providing the appropriate functionality.

3.3 Service Trading

Transparent information and service access independently from the user's location is an important requirement for mobile computing. If a mobile moves between different networks the services being available may change depending on its current location. Additionally, if the mobile provides services to other computers the topology is changing, too. This must be supported by an extended arbitration of services. The interaction of components and protocols to support service provision and transparent use is depicted in Figure 4.

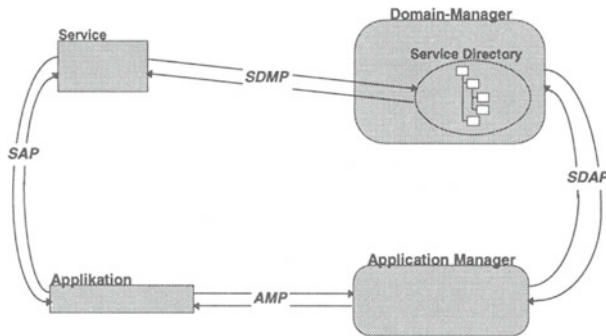


Figure 4 Service Trading Protocols

In a so called *Service Directory* the **domain manager** holds the services that are available in his domain. He knows which services are available on each station, which access rights are needed and which stations and applications currently use the services.

The **application manager** holds information about the services that are needed to perform the applications. He has to negotiate the use of services with the domain manager based on information specified in application profiles. The application manager decides which service is used if more than one suitable service is available. Therefore the trading functionality is part of the application manager. The service handover to similar services in cases of disconnection or moves between domain is also done by the application manager.

The **station manager** provides a transparent connection between application and service provider. If a new service is to be used, a connection to the new service endpoint is established by the station manager (triggered by the application manager). This change is transparent to the application unless specified otherwise in the application profile.

The registration and deregistration of services in the service Directory is done by the **Service Directory Management Protocol (SDMP)**. The application manager queries the service directory using the **Service Directory Access Protocol (SDAP)**. The SDAP distinguishes between queries for one service and for a list of services. The server-specific **Service Access Protocol (SAP)** handles the communication between application and service. The **Application Management Protocol (AMP)** is used for the information exchange between application and application manager. It covers queries for services as well as service handover.

4 SERVICE MOBILITY AND SERVICE HANDOVER

Mobile users want to use the same services regardless of their current location. The need for transparent access to services implies service mobility (LaKr 94). Several approaches are possible:

- The simplest form is to keep all connections to services open when the mobile computer moves. This can be realized by transparent packet routing to the current location (e.g., Mobile-IP). There is however a lack of performance because of rerouting.
- Another possibility is to transfer the used services and data or the entire working environment to the new location. The problems here are heterogeneity of hard- and software as well as concurrency and consistency. Services had to be replicated and the data consistency has to be maintained.
- The third approach is to use local services. If there is a similar service at the new location, this service can be used. To keep this transparent to the user, a so called service-handover is needed. This not only maps the connection from the old to the new server, but also transfers context information that includes explicit knowledge of the users current location.

This section describes how MOBI-DICK provides a powerful support for service mobility. The scenario to be discussed is shown in Figure 5. There is one domain that consists of two subnets. A mobile user is connected to subnet 1 and uses locally available services, e.g., database query, print service. He then moves to subnet 2 without quitting his application. For performance reasons the local services of subnet 2 should be used, if they are appropriate. In this case a service handover is necessary. We now describe the steps to perform this handover in detail.

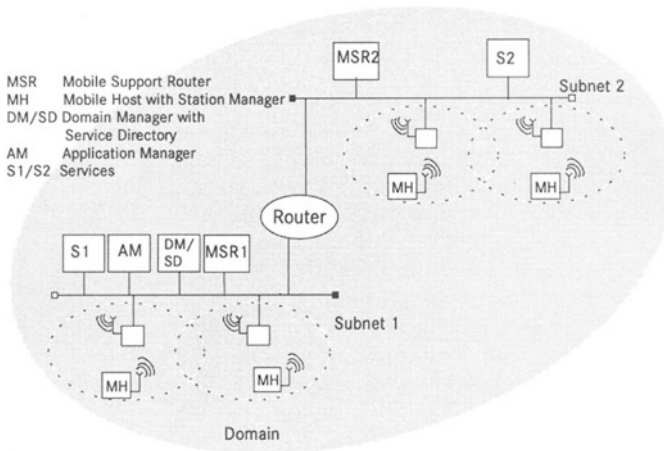


Figure 5 Scenario with one domain and two subnets.

Registration in Subnet 1

First, the mobile computer/user (MH) has to register (see upper part of Figure 6) at the Mobile Support Router of subnet 1 (MSR1). For simplicity reasons, it is assumed that MSR1 also is the Home agent (HA) of the MH. Then the MH acting as a MOBI-DICK station registers at the domain manager (DM). With this, the domain manager gets the IP address of the Home

Agent (i.e., MSR1). The domain manager then asks the MSR1 for the current location of MH - here : subnet 1.

Service Trading in subnet 1

Then the services that are needed by the applications have to be provided (see upper part of Figure 7). The application manager (AM) sends a query to the service directory (SD) and selects the appropriate services. In this scenario the application uses only local services in subnet 1.

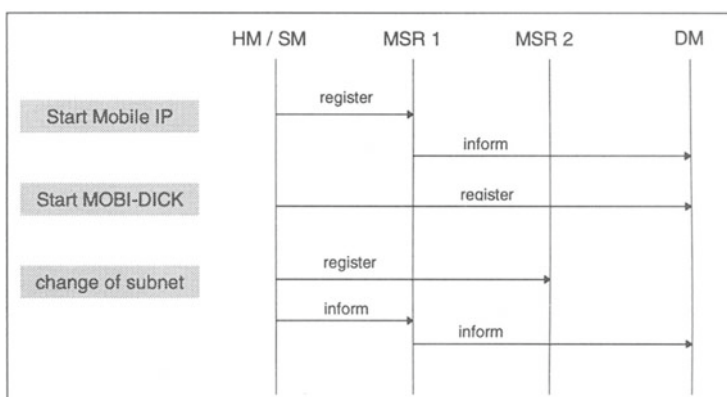


Figure 6 Protocols between Mobile Host, MSR, DM.

Migration from subnet 1 to subnet 2

The mobile computer (MH) then moves from subnet 1 to subnet 2 (see lower part of Figure 7). Please note that both subnets belong to the same domain. In subnet 2 the MH registers at the Mobile Support Router MSR2, which acts as Foreign Agent. Furthermore, the MH informs his Home Agent MSR1 about his new location: subnet 2. To use the local services in subnet 2, the current location of the MH must be explicitly known. For this, the Home Agent MSR1 sends a SNMP Message to inform the domain manager about the new location of MH. Additionally the Application Manager (AM) is informed. Now the information about the new location can be used to handover the service from subnet 1 to a service in subnet 2.

Service Handover

After being informed about the new location of MH, the application manager sends a new query to the service directory and gets a new list of services with respect of the new location. The actual service handover can be performed automatically or user driven. The AM triggers Service 1 to reconnect to a new service: Service 2. After the necessary context information is transmitted, the application can use Service 2.

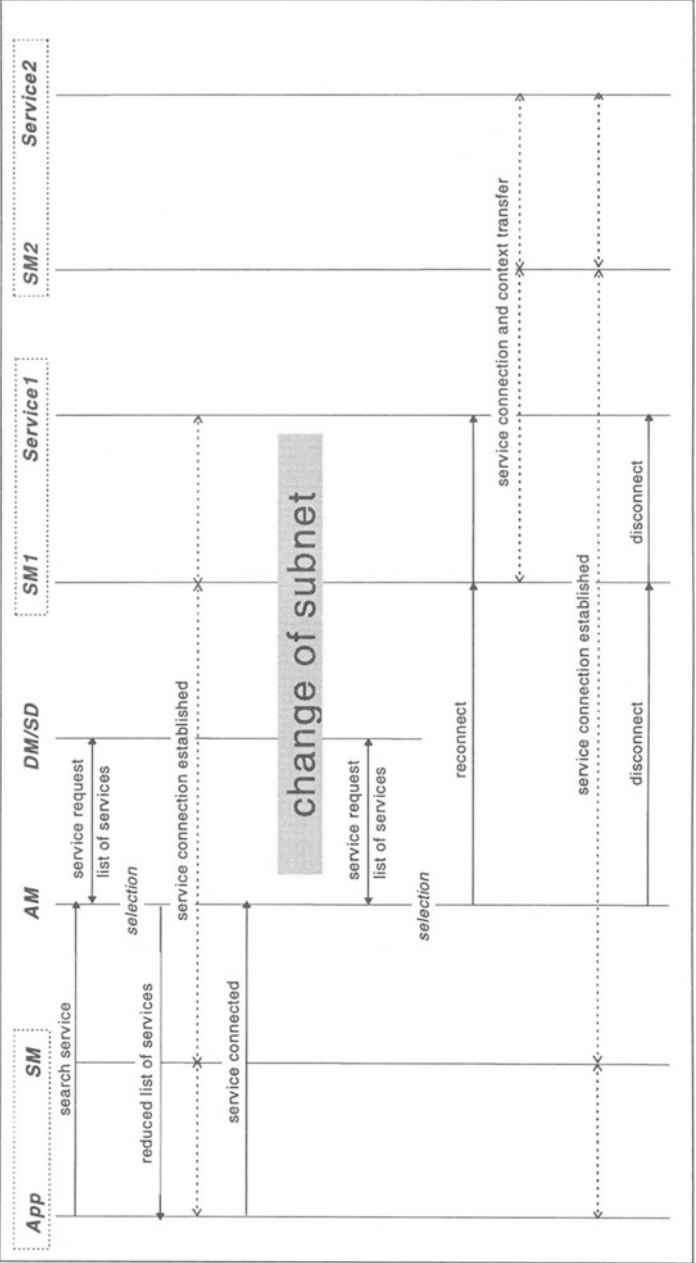


Figure 7 Service trading and handover protocols.

5 TEST SYSTEM AND CONCLUSIONS

At our Research Center we are currently implementing a test system based on the MOBI-DICK-architecture. The current system uses the remote procedure calls (RPCs) of the Distributed Computing Environment (DCE) of OSF (Open System Foundation) (OSF92) for the communication between the stations. The RPC is based on TCP or UDP. As wireless communication systems we use the PCMCIA-based radio LANs from DASA (Meshnet2) and Xircom (Netwave) and GSM for wide area communications.

The implementation is done on MS-Windows PCs under Smalltalk (Visual Works) for the higher Layers (station manager and domain manager) and Microsoft Visual C++ for the communication interface. For the realization of the service mobility RoamAbout Mobile-IP from DEC is used. In a first step the scenario described above with a mobile host moving between two subnets within one domain and the active provision of a local printer was implemented. This is shown in Figure 8. We tested the functionality of our system within a prototypical application to support customer services at Mercedes-Benz dealers.

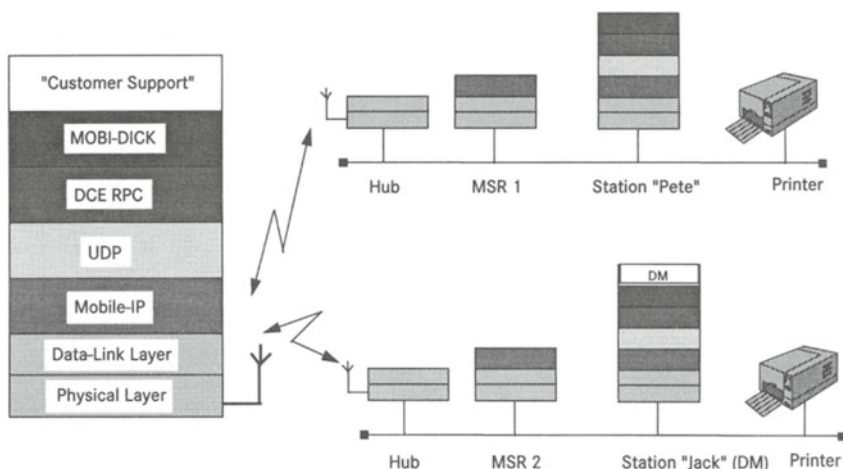


Figure 8 Layers of the test system

The systems works fine in the local area environment. However we found some problems using the standard RPC of DCE. The main problems are intermittent connectivity and the relatively high overhead for small bandwidth channels like GSM. Therefore we are looking at modifications of the RPC towards mobile RPC and indirect concepts with generic protocols for the wireless communication part.

Additionally, more work has to be done in the field of mobile service trading. Up to now we use a proprietary implementation for the service directory. In the future we will investigate how far CDS and GDS of DCE, CORBA of OMG, and the concepts of the Intelligent Network (IN) can be used for mobile service trading.

6 REFERENCES

- [Dieh 95] Diehl, N.: Field Support using Mobile Computing. *Proc. of European Conference on Product and Process Modelling in the Building Industry*, Dresden 1994 Balkema Publishers, pp. 325 - 332, 1995.
- [DGH 94] Diehl, N.; Grill, D; Held, A.; Kroh, R.: System-Platform for Mobile Computing. Personal Indoor am Mobile Radio Conference, *PIMRC'94*, Den Haag, pp. 1353 - 1358, 1994
- [DiHe 94] Diehl, N.; Held, A.: A System-Platform for Mobile Computing Applications. *Proc.s of MOBIDATA-Workshop (IEEE; NSF)*, Rutgers Univ. NJ, 1994
- [ImBa 94] T. Imielinski, B.R. Badrinath: Mobile Wireless Computing; *Communications of the ACM*; Vol 37, No 10, Oct. 1994.
- [Weis 93] M. Weiser: Some Computer Science Issues Related to Ubiquitous Computing; *Communications of the ACM*; Vol 36, No. 7, 1993.
- [Saty 93] M. Satyanaryanan et. al.: Experience with Disconnected Operation in a Mobile Computing Environment; *Proc. of 1993 USENIX Symp. on Mobile and Location-Independent Computing*, 1993.
- [IPMS 95] *IP-Mobility Support*; IETF Internet Draft, 1995.
- [ScKü 95] A. Schill, S. Kümmel: Design and implementation of a support platform for distributed mobile computing, *Distr. Syst. Engng.* 2, pp. 128 - 141, 1995.
- [LaKr94] R. Jain, N. Krishnakumar: Network Support for Personal Information Services to PCS Users, *Proc. of Networks for Personal Communications (MPC) '94*, 1994
- [OSF 92] Open Software Foundation: Introduction to OSF DCE, *Prentice Hall*, 1992.