Service Architecture for an Object-Oriented Next Generation Profile Register

Andreas Diehl [1], Udo Bleimann [2], Woldemar Fuhrmann [2], Paul Reynolds [3], Stephan Rupp [1]
[1]Alcatel SEL AG, Stuttgart, Germany
[2] University of Applied Sciences Darmstadt, Germany
[3] France Telecom Research and Development, United Kingdom
email: andreas.ad.diehl@alcatel.de

Abstract

This paper describes a new approach for Object-Oriented (OO) Service Handling, Service Deployment, Service Architecture and the corresponding Data Model Interworking of a Next Generation Profile Register (NGPR). This Profile Register should fulfill the specific service requirements of mobile telecommunication network operators that are related to the efficient creation and installation of new services, service access and service registration.

Keywords

UMTS, GSM, Home Location Register (HLR), Home Subscriber Server (HSS), Next Generation Profile Register (NGPR), Next Generation Networks, Subscriber Profile, User Profile, General User Profile, Service Framework, Service Deployment

1. Introduction

In current mobile telecommunication networks, databases and subscriber data are distributed among multiple locations and network nodes. For example, in UMTS Release 5 (3GPP TS 21series, 2005) (3GPP TS 22series, 2005) (3GPP TS 23series, 2005), the user profile is stored in one or several Home Subscriber Servers (HSS) consisting of IP Multimedia Home Subscriber Servers (IMHSS) and Home Location Registers (HLR). The concept of a Next Generation Profile Register (NGPR) presented in this paper proposes a novel object oriented approach using a logically-centralized register for all kinds of subscriber-related data (Lopez-Aladros et al, 2003) (Rupp et al, 2004/1) (Rupp et al, 2004/2). This data can be accessed for various purposes in a generic way, directly via next generation network nodes, traditional GSM (3GPP TS 41series, 2005) (3GPP TS 42series, 2005) (3GPP TS 43series, 2005) and UMTS network nodes. The previous network nodes may use gateway functionalities for interoperability. The NGPR covers several fields, e.g. authorizing the access to specific bearer services, determining user location, billing, Customer Relationship Management (CRM), Data Mining and Identity Management. It is also supported by open access protocols, generic application and service integration (Diehl et al, 2004). Due to the difficulty of evaluating and foreseeing new applications and services, a flexible data structure and service creation is supported. This paper presents an approach for an Object-Oriented (OO) service framework utilizing a flexible OO-data model.
In addition to the service model the data model is also proposed to be object-oriented. Object-Oriented Programming (OO-P) defines objects that encapsulate data and operations on the data. Objects allow the reusability of code and dedicated interface definitions are used for interoperation. Apart from supporting application logic, modern object-oriented concepts

additionally support persistent storage capabilities. OO-Databases (OO-DB) offer the required functionalities for handling persistent objects (object storage and retrieval, transaction management, security, etc.). Although there is a clear separation between persistent and runtime data in the application, this data becomes ambiguous when modelling the data using a particular modelling language.
Persistent data and run-time data can be modelled using the same language. The need to translate persistent data objects into relations or other data model structures is eliminated. Object references are used to identify and address data objects, and the search process can directly use the existing object tree to speed up the search process.

In the following the network of a single operator is considered and issues like problems of multiple operators accessing one NGPR are excluded.

2. The Proposed Framework of a Next Generation Profile Register

Figure 1 gives an overview of the NGPR. In addition to multiple Profile Databases there is a Localization Database, which is used to select the Profile DB and thus the subscriber profile.

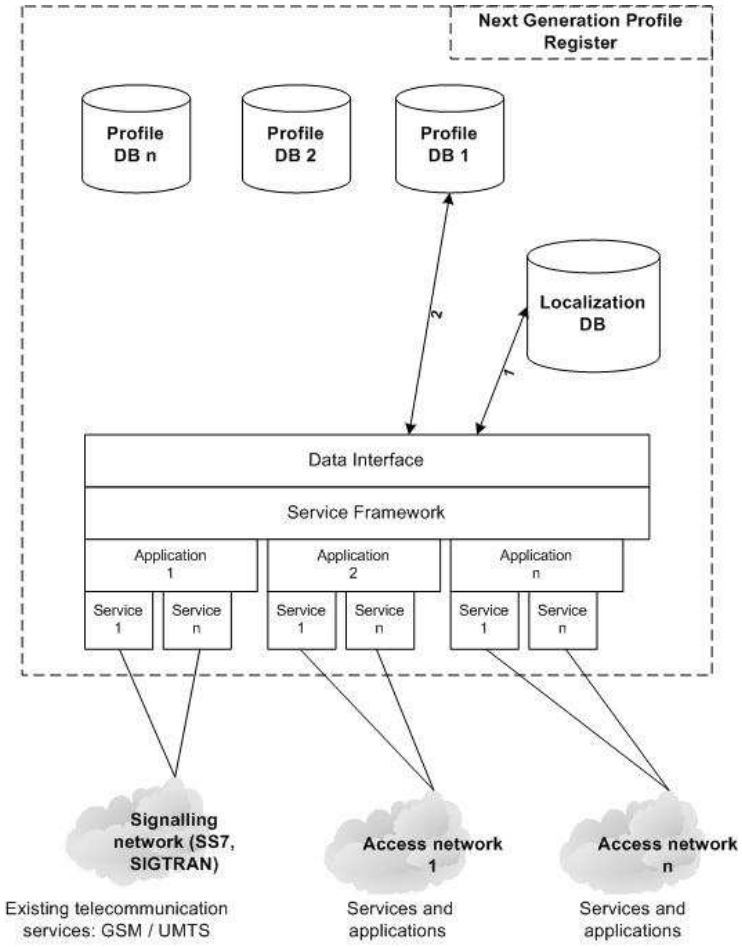

Figure 1 - Overview of the Next Generation Profile Register

The subscriber profile is identified by a profile key, which is stored in the Localization Database. Fast data access is guaranteed by the mechanism of selecting the Profile DB.

The Service Framework will enable an efficient deployment of services allowing any type of data sets. This framework supports legacy or new services, which are deployed by a well-defined process and which are registered without the need of a system restart ("plug-in").

The Data Interface aims to provide various networks common access to the general, integrated profile data. The offered services may also be reused for composing new services, for example, a wireless-local area network (WLAN) service may use parts of the offered HLR services for its authentication and authorisation process. These service-oriented concepts are well known, for example, from Web Services (Bellwood, 2002) (Christensen et al, 2001) (Fallside et al, 2001) (Glass, 2002) (Mitra et al, 2003).

3. Service Deployment for a NG Profile Register

The data models used for traditional HLRs are often provider specialised and make the integration of the various subscriber and business data of a network operator or service provider very difficult. Service registration and deployment is performed manually by updating the software of every single HLR area. Services provided by the NGPR are centralized and therefore more efficient to administrate. OO-P offers the possibility of object-based development, object-based updates and information hiding. Furthermore OO-P provides new possibilities for data modelling using object trees.

Object-Persistence can be achieved by using an appropriate standard, for example Java Data Objects (JDO) (Russel, 2001). Other specialised vendor-dependent database interfaces are possible. According to the database system and programming language, database schemes or enhanced classes with descriptors define the object structure. Manipulation operations (e. g. delete, update etc.) of a persistent class can be achieved by a schema evolution. A database schema compiler can be used for automatic generation of the data model description. The compiling process for a schema can be integrated in the deployment process of a new service.

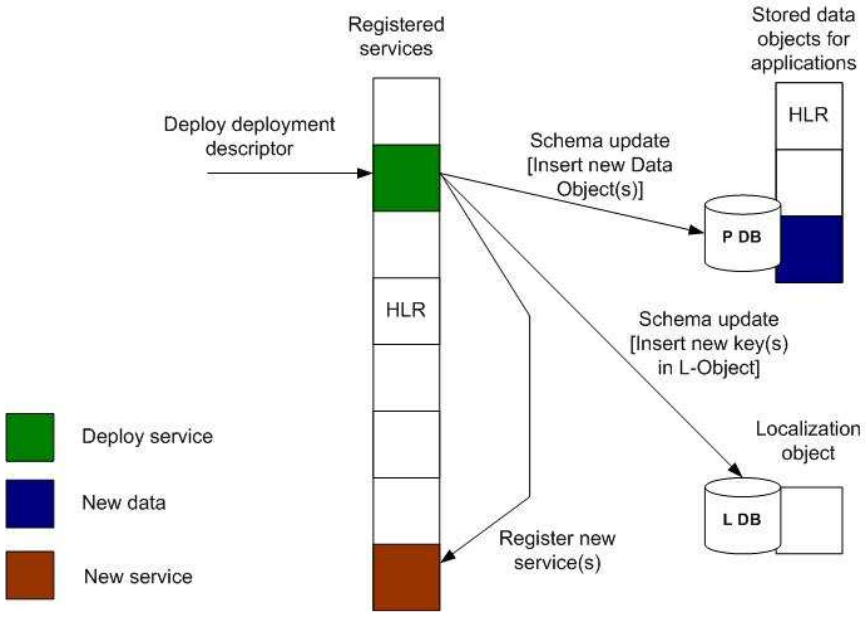

Figure 2 - Service deployment

Figure 2 shows a deployment process, which updates the description of the NGPR data model (Persistence Descriptor) and registers the new service (Service Descriptor) in the Service Framework. This is followed by a database schema evolution in the Profile Database to support the persistent data of the registered service. If the new data also requires one or more new profile key(s), the Localization Database should also be updated. This is necessary, because the Localization Database provides a reference to the Profile Database via the profile key. If the profile is extended, a load process needs to be considered, which transfers the actual profile data from the legacy systems of the specific access network to the NGPR: After the load process the service is completely registered and can be set online.

This service deployment process offers an efficient and fast mechanism to introduce new services and the corresponding data structures.

4. Data Model for a NG Profile Register

The data model should be independent of specific network or service architectures. An example of an object-oriented Profile Data Model is illustrated in Figure 3. The superset object (CommonProfileData class, refer to Figure 3) of the data model identifies the complete set of subscriber-related data combined in the subscriber's profile. This profile consists of different data sets associated with the data storage entities of the various access networks.

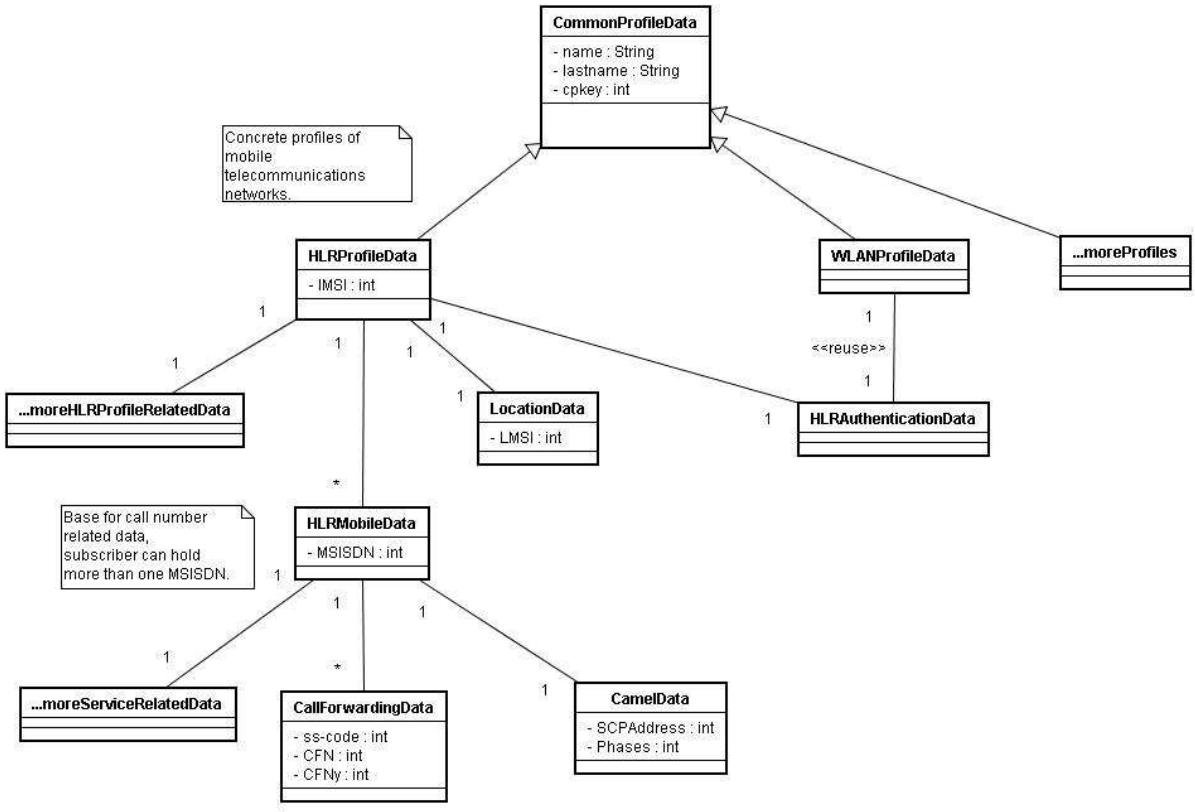

Figure 3 - Example of an object-oriented profile data model

For new applications, a subscriber profile is identified by the key of its complete profile. Therefore, run time services dedicated to the applications can access and integrate the

complete data sets related to the subscriber. For legacy applications, the existing keys of the corresponding data storage entities may be used. As an illustration, a HLR data model is considered as an example for the NGPR, which describes persistent mobile subscriber data, such as location, basic telecommunication services, supplementary services etc.

In a HLR two primary keys are defined for database access, the International Mobile Subscriber Identity (IMSI) and the Mobile Station ISDN Number (MSISDN). Conventional GSM services (3GPP TS 42 series, 2005) (3GPP TS 09.02, 1998) use the IMSI for access to the HLR profile data and the MSISDN as a key for the HLR mobile data. Billing services, for example, may access the whole profile by using the complete profile key (cpKey, refer to Figure 3). The HLR profile inherits common information about the subscriber, which is stored in the superset object.

HLR data, for example location or authentication data, can be associated or inherited by other storage entities, which may belong to different access networks. This means that data can be reused between the different data storage entities.

Profile data of a subscriber from different access networks is always identified by the subscriber profile key. Therefore, the superset object references to one or more specific profiles of the data storage entities of an access network. The specific data storage entity contains the specific profile-related data. Data storage entities other than HLR can be easily "plugged in" via inheritance of the root object. After the integration of a new data storage entity the database schema of the actual data model is thus updated. A new database storage entity may belong to another access network, such as a WLAN, or it may provide different data formats, such as video, audio, Instant Multimedia Messages etc.

5. Services and Application Model for a NG Profile Register

The service and application model supports a service-oriented architecture. Application and administration services are distinguished. For example, the deployment process can be seen as an administration service offering deploy and undeploy functionalities (See Section 2). Other administrative services include, load services used for inserting new service objects or for copying data from legacy network nodes, etc.

Application services may be further sub-divided into run-time and provisioning services; services in general offer create, update, modify or delete operations related to user interfaces on the subscriber profile. The HLR services for GSM and UMTS are, for example, part of the HLR application. Other applications may include different access networks, or new service platforms. Specific HLR application services include, for example, the legacy signalling operations of the Mobile Application Part (MAP) (3GPP TS 09.02, 1998), such as UpdateLocation, CancelLocation, SendAuthenticationInfo etc. Concrete HLR application services use a concrete factory to build the concrete data object (refer to classes in Figure 4). The concrete factory class always inherits from a super application factory class.

This architecture strictly separates the services together with their building processes from the persistent data (refer to Figure 4). The service logic is encapsulated in the specific service object and only the interface of the service object needs to be advertised to a broker or trader function.

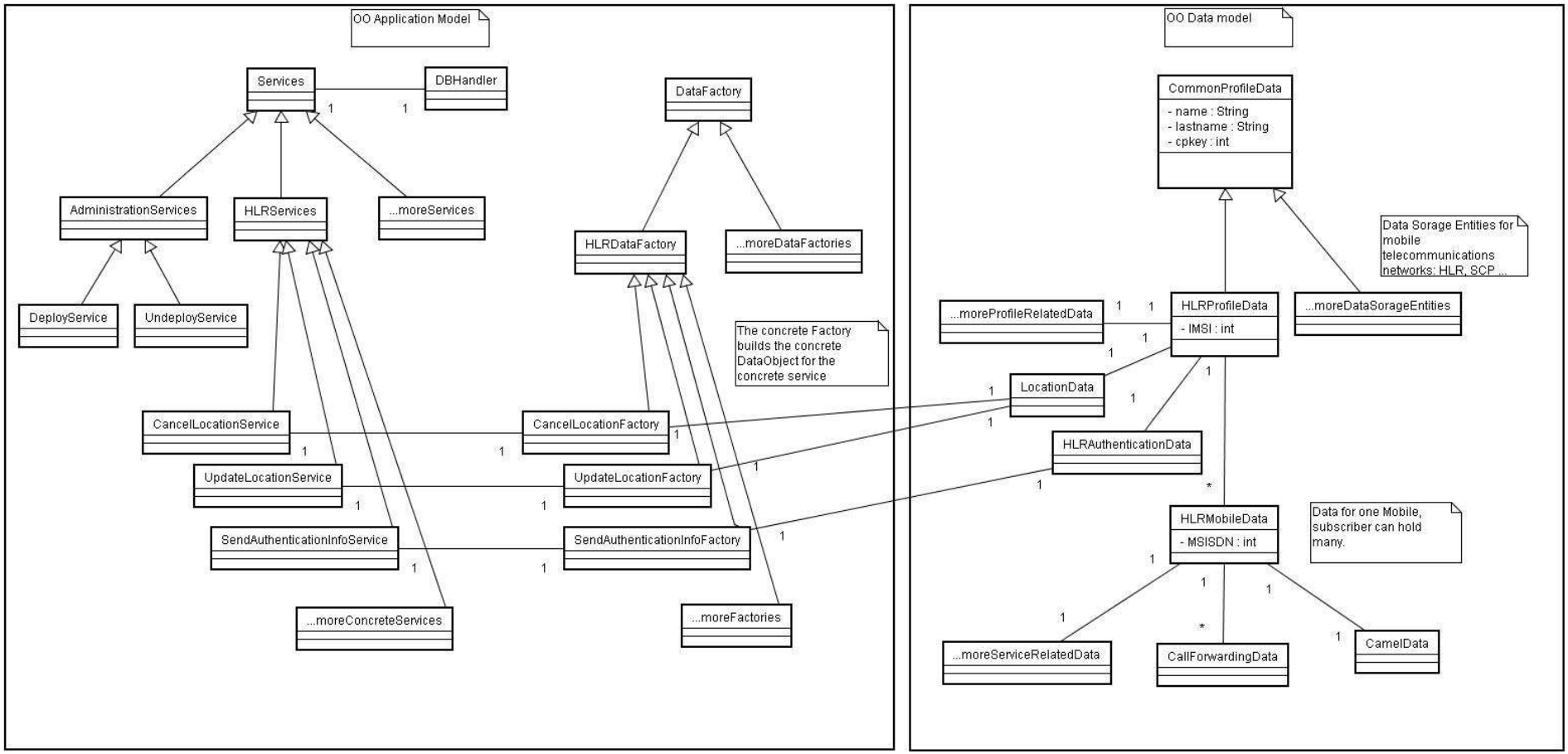

Figure 4 - Example of an object-oriented service and data model

"Get" and "set" methods of the concrete data object can be used by the concrete factory for data object interaction. The concrete factory, depending on the operation type, is responsible for building the object. A database handler is used to integrate the built objects in the Profile Model and objects that have been set as persistent can be stored in the database.

For example, the object interoperation for an update operation could consist of the following steps:

1. An update service class is requested
2. The service class retrieves the selected data object using the database handler. (The service class identifies the location of the concrete key)
3. The service class calls a specific factory update method with the data object and request type as parameters. This builds the update object.
4. The service class uses the database handler for setting the update object as persistent.
5. The service class returns from the update operation.

The service and application model is protocol independent. For a sample implementation SOAP protocol (Mitra N. et al, 2003) and Web Services are considered. SOAP based Web Services are a common supported standard for Service-Oriented Architectures (SOA). SOA provide an easy service registry (Bellwood, 2002) (Christensen et al, 2001) and service access. This technology opens the door for telecommunication providers to integrate future applications (e.g. CRM, Data Mining, Identity Management) with traditional profile data. Interoperability and data integration of traditional profiles is the future field of our research.

6. Conclusions

The concept of a NGPR describes a logically-centralized object-oriented subscriber database, which allows the integration of all subscriber related data in a subscriber profile. This approach provides automatic deployment of new and modified services. There is only one common profile of subscriber related data supporting the efficient integration of multiple applications. The data model is modular and data objects form the building blocks that can be reused in different services and applications. The access to data objects is fast using state of the art OO-DB technologies and all kinds of data are supported. Legacy systems and their data models can be integrated using appropriate gateway adaptation functions.

7. References

3GPP TS 41 series (2005), "GSM only Requirements", http://www.3gpp.org/ftp/Specs/html-info/41-series.htm, (Accessed 25 March 2005), Valbonne; France.

3GPP TS 42 series (2005), "GSM only Service Aspects (Stage1)", http://www.3gpp.org/ftp/Specs/html-info/42-series.htm, (Accessed 25 March 2005), Valbonne; France.

3GPP TS 43 series (2005), "GSM only Technical Realization (Stage2)", http://www.3gpp.org/ftp/Specs/html-info/43-series.htm, (Accessed 25 March 2005), Valbonne; France.

3GPP TS 21series (2005), "3G GSM R99 and later Requirements", http://www.3gpp.org/ftp/Specs/html-info/21-series.htm, (Accessed 25 March 2005), Valbonne, France.

3GPP TS 22series (2005), "3G GSM R99 and later Service Aspects (Stage1)", http://www.3gpp.org/ftp/Specs/html-info/22-series.htm, (Accessed 25 March 2005), Valbonne, France

3GPP TS 23series (2005), “3G GSM R99 and later Technical Realization (Stage2)”, http://www.3gpp.org/ftp/Specs/html-info/23-series.htm, (Accessed 25 March 2005), Valbonne, France.

3GPP TS 09.02 (1998), “Mobile Application Part Specification (MAP)”, http://www.3gpp.org/ftp/Specs/html-info/0902.htm, (Accessed 25 March 2005), Valbonne, France.

Bellwood, T. (2002), “UDDI Version 2.0.4 API Specification”, OASIS, http://uddi.org/pubs/ProgrammersAPI-V2.04-Published-20020719.pdf, (Accessed 25 March 2005)

Christensen, E., Curbera F., Meredith G. and Weerawarana S (2001),“Web Services Definition Language (WSDL)”, W3C Note http://www.w3.org/TR/wsdl, (Accessed 25 March 2005)

Diehl, A., Fuhrmann W. and Siegmund G.(2004), “MAP-SOAP Interworking in Future Service-Oriented Wireless Networks”, *International Network Conference*, Plymouth.

Fallside D. (2001), “XML-Schema”, *W3C Recommendation Part 0*: Primer, http://www.w3.org/TR/xmlschema-0/, (Accessed 25 March 2005)

Glass, G. (2002), *Web Services: Building Blocks for Distributed Systems*, Prentice Hall PTR.

Lopez Aladros, R., Banet, F.-J., Rupp, S., Winter, E. (2003), “Restructuring the Telecommunication Networks”, *Working Conference HET-NETs’03*, Ilkley.

Mitra N. (2003), “Simple Object Access Protocol Specification Ver. 1.2”, *W3C Recommendation.*
Part 0: Primer, http://www.w3.org/TR/2003/REC-soap12-part0-20030624/. (Accessed 25 March 2005)
Part 1: Gudgin M., Hadley M., Mendelsohn N., Moreau J and Nielsen H., ”Messaging Framework”, http://www.w3.org/TR/soap12-part1/, (Accessed 25 March 2005)
Part 2: Haas H, Hurley O., Karmarkar A., Mischkinsky J., Jones M., Thompson L. and Martin R., ”Adjuncts”, http://www.w3.org/TR/2003/REC-soap12-part2-20030624/ (Accessed 25 March 2005), “Specification Assertions and Test Collection“, http://www.w3.org/TR/2003/RECsoap12-testcollection-20030624/, (Accessed 25 March 2005).

Rupp S., Siegmund G., Lopez-Aladros R. and Banet F. (2004/1), “Flexinet – A Network Service Architecture”, *Journal of the Communications Network*, Jan-March 2004.

Rupp S., Lopez-Aladros R., Banet F. and Siegmund G. (2004/2) “Flexible universal networks - a new approach to telecommunication services”, *The 8th World Multi-Conference on Systemics*, Cybernetics and Informatics, Orlando 2004.

Russell, C. (2001) “Java Data Objects Version 1.0, Sun Microsystems”, *Java Community Process (JCP)*, http://www.jcp.org/aboutJava/communityprocess/first/jsr012/, (Accessed 25 March 2005), Palo Alto, California U.S.A.