A Flexible Object-Oriented Data Model for a Centralized Subscriber Profile Database

Andreas Diehl, Alcatel SEL AG, Stuttgart, Germany, Franz-Josef Banet, Alcatel SEL AG, Stuttgart, Germany, Gerd Siegmund, Technik der Netze, Stuttgart, Germany, Stephan Rupp, Alcatel SEL AG Stuttgart, Germany and Woldemar Fuhrmann, University of Applied Sciences Darmstadt, Germany

***Abstract*—A centralized subscriber profile database promises cost savings and other advantages for operators of GSM and UMTS networks. Of course having all subscriber profiles in a single location (and surely a mirror location) reduces the complexity of today's geographically distributed profile storage. But to collect all subscriber profiles and to shift them into a big central database is not sufficient. In addition it is necessary to combine them in a common data model. This paper describes an object-oriented data model for a centralized subscriber profile database and its extensibility. Extensibility of the data model is necessary for enlargement with new applications and also for enlargement with other access networks like WLAN or Bluetooth that need add-ons to the subscriber profiles.**

Object-Oriented Data Model, Subscriber Profile, UMTS, WLAN

I. Introduction

TODAY profiles of the same subscriber are existent in different network entities of a single operator domain, as there are for example the Home Location Register (HLR), the Home Subscriber Server (HSS), the Service Control Point (SCP) and others. Each of these entities is the owner of its profile data. There are of course some redundancies if we look at a certain subscriber and his profiles in those network entities. Other parts of these profiles do not overlap; this means they are only dedicated to an usage in a single type of the network entity, in a HLR for example [2].

Doing so is possible because standardization bodies like 3GPP (3rd Generation Partnership Project) only define the functionalities of a network entity and its external interfaces. The internal representation of a subscriber profile inside a network entity or application and the associated data model are not specified in standards. Each HLR manufacturer for example can define an own data model for the HLR subscriber profiles. Advantages of the centralized approach are the simplification of administration, the storage of all kinds of data and the support of services for a complete subscriber profile. These services are Billing, Customer Relationship Management (CRM), Data Mining and Identity Management.

This paper describes the concept for a flexible, extensible data model of a logically-centralized subscriber profile database and excludes performance aspects, service [7] and session handling.

Manuscript received January 26, 2005. This work is co-funded by the European Commission under the framework of IST-FP6-1-507646-STP Project, FLEXINET - Flexible Network and Gateways Architecture for Enhanced Access Network Services and Applications.

Andreas Diehl is with Alcatel SEL AG, D-70435 Stuttgart, Germany (phone: +49 (0)711 821 44230, e-mail: andreas.ad.diehl@alcatel.de).

Franz-Josef Banet is with Alcatel SEL AG, D-70435 Stuttgart, Germany (e-mail: fj.banet@alcatel.de).

Gerd Siegmund is with Technik der Netze, D-70435 Stuttgart, Germany (e-mail: stz745@stw.de)

Stephan Rupp is with Alcatel SEL AG, D-70435 Stuttgart, Germany (e-mail: s.rupp@alcatel.de)

Woldemar Fuhrmann is with University of Applied Sciences Darmstadt, D-64295 Darmstadt, Germany (e-mail: w.fuhrmann@fbi.fh-darmstadt.de)

II. Aggregation of Subscriber Profiles

The data model for the Common Profile, which we define as the superset of the several subscriber profiles, must represent all these subscriber profiles. As shown in Figure 1 the subscriber profiles of the legacy network entities move into the Common Profile.

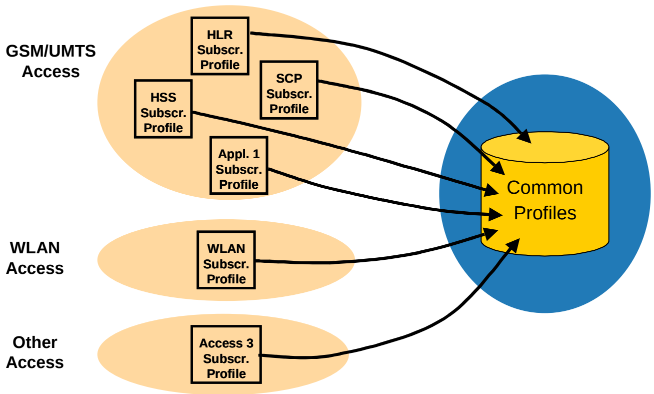

Fig. 1. Subscriber Profile move into the Common Profile.

Once this is done a legacy application like the HLR must no longer store subscriber profiles in its own database. This means a new-fashioned legacy application, which we call a Common Profile capable application, can mainly be reduced to its application logic [1]. A Common Profile capable legacy application is representing the same functionality as the legacy application, but is working with the centralized subscriber profile database. For a job the needed profile is fetched from the centralized subscriber profile database and forgotten after job termination. Of course all Common Profile capable legacy applications must recognize their subscriber profile in the Common Profile.

Other access types than GSM or UMTS like WLAN or Bluetooth might require additional profile data for a subscriber who is already subscribed to GSM or UMTS. These profile data are also included in the Common Profile. The aim is to describe within the Common Profile the subscriber with all his attributes, including all possible access types and all subscribed services. Starting point for our definition of a Common Profile is the main subscriber profile in GSM and UMTS networks, the HLR subscriber profile. Many parameters in the HLR subscriber profile can be reused for other purposes, e.g. for authentication procedures in a WLAN [3].

Once a centralized subscriber profile database exists, all new introduced applications and network entities can be Common Profile capable ones and then profit from already existing subscriber data in the Common Profile. Additional parameters might be necessary and can be added to the Common Profile. Of course legacy applications or network entities, which are still working in the traditional way, e.g. a legacy HLR, can coexist in a network with Common Profile capable applications until they will be replaced by a successor. Figure 2 shows the coexistence of legacy network entities with new applications and network entities that use the centralized subscriber profile database.

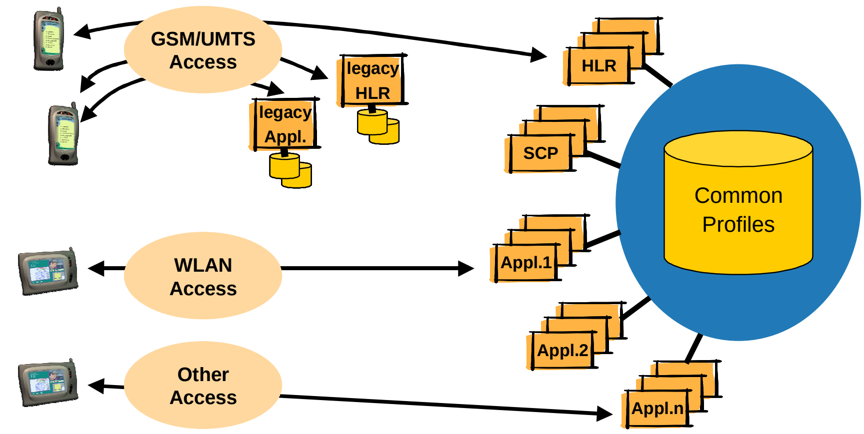

Fig. 2. Coexistence of legacy network entities with Common Profile capable applications.

III. WHY OBJECT ORIENTED ?

The preliminary data model of the centralized subscriber profile database is assumed to be object-oriented. Modern object-oriented concepts include not only application logic, but also persistent storage support. Object orientation offers several advantages according to data modelling. An object itself is independently organized and a set of objects can be encapsulated as one component. Components can be related to their functionality or network. Data access is guaranteed by function calls. Interface definitions provide interoperability to the independently organized data objects. Data interworking and design can be integrated in the application design by the use of the same model, without any translations of the data model. For example object-relational mappings are not necessary. Of course applications still separate between persistent data and run-time application logic.

Objects are using references for object-access, these references can be reused by the object-oriented database systems (OO-DB) to automatically optimize the search process. For profile groups support object creation can be predefined by the application.

The object-oriented data model offers the common data-types which enable more than subscriber profile storage, for example multimedia storage.

OO-DBs using standards (e.g. Java Data Objects [4]) or specialized database dependent interfaces for object persistence. According to database solutions and programming language database schemes, enhanced classes or descriptors define the object structure. The schemes can be generated by the object model and updated for new data integration. This feature offers a maximum on flexibility for data manipulation. State of the art OO-DBs support all required base functionalities known from relational databases.

IV. DATA MODEL OF A CENTRALIZED SUBSCRIBER PROFILE DATABASE

The base object in Figure 6 (Object CommonProfileData) is considered as one object, which identifies the Common Profile of the subscriber/user by a common profile key (cpKey). The Common Profile contains superset information about the subscriber, for example the name and address. Common information is provided to the concrete profile, because each concrete profile has to inherit from the Common Profile. So the complete profile consists of different data separated by the data of the data storage entities of the access network. These data storage entities contain the concrete data model as used in the legacy systems (e.g. in a HLR). Subscribers of a concrete network can be identified for future applications by the key of the whole profile (cpKey). Existing applications use the existing keys to access the profile.

The data model provides flexibility and a mechanism for modification operations of concrete sub-profiles dedicated to a specific access network. The functionality of adding and deleting a concrete profile is a strict requirement, because access networks and their data may change. Therefore the sub-profiles are encapsulated in components and separated by profile requirements of the access network. Add and delete operations can be supported by manipulating the concrete component.

Figure 3 shows an overview of the Common Profile data and the concrete profile components HLR, WLAN and SCP. Each object of one component should be reusable, if the data is not specific to the concrete access network. As an example the authentication data of the HLR can be reused by the WLAN data. Concerning this the GSM Subscriber Identity Modul (SIM) can be adopted by card readers to a WLAN capable personal computer.

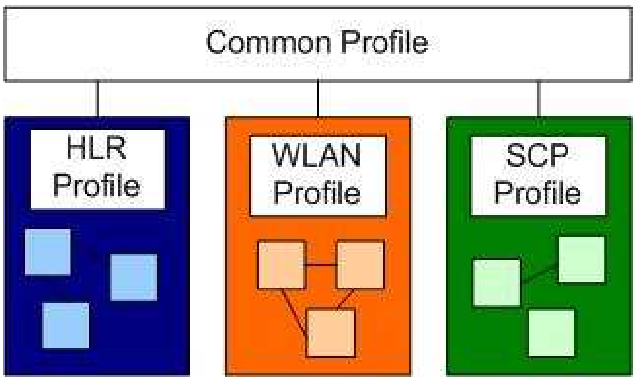

Fig. 3 Seperated concrete profiles

Then authentication can be achieved over Enhanced Authentication Protocol (EAP/SIM) [6]. Consequence would be an association between the WLAN authentication object and the HLR authentication object.

After this modification a link program has to build all references between the existing database objects (refer to Figure 4). This can be done, because each concrete profile holds the inherited common profile key. This key is used for linking objects between the concrete profiles.

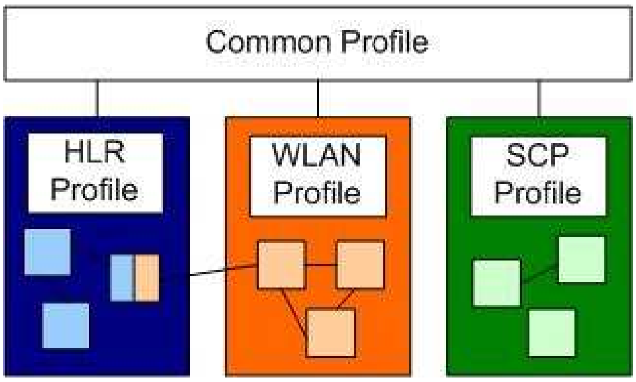

Fig. 4. WLAN profile reuses HLR profile object

To support manipulation operations between linked objects of a component the link states have to be stored, for example if the provider likes to delete the HLR profile of the GSM access network (as shown in Figure 5).

The link state between components has to be stored either in embedded object parameters or an external meta-description of the underlying class diagram. The use of embedded object parameters would have the advantage to proof run-time conditions of an instance. This would offer the more detailed possibility of object manipulation. The meta-description has the advantage of clear separation between profile data and administration data. In both cases an application has to proof and perform the manipulation of the concrete profile. Update or delete operations have to be restricted, because changes of concrete objects of one access network may influence concrete objects of other access networks (refer to HLR link in Figure 5). Restrictions and rules have to be defined. For the database the schema of the new class structure has to be generated and a schema evolution has to be committed to the database.

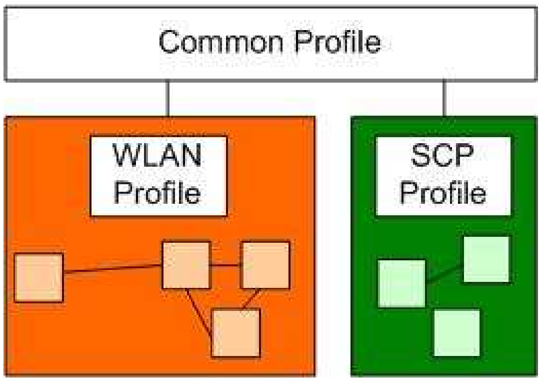

Fig. 5. HLR profile was deleted, WLAN profile still holds the former HLR object

As an example the HLR data model is shown on a high level perspective in Figure 6. For a HLR there are two primary keys, which are used for database access: International Mobile Subscriber Identity (IMSI) and Mobile Station ISDN Number (MSISDN).

The traditional HLR services of the legacy applications, e.g. location updates, are able to use the well-defined keys. The IMSI is used as key for the common subscriber data (HLRProfileData) of the HLR, the MSISDN is used as key for the concrete mobile data. Billing procedures can use the whole profile by accessing data via the common profile key (cpKey). By definition are IMSI dependent data for example location data and authentication data. The class HLRMobileData stores all call number dependent data, for example data about call forwarding or Customised Application for Mobile network Enhanced Logic (CAMEL).

The HLR profile supports more than one mobile device per subscriber. Therefore the reference to the mobile data is associated into a one to many relationship.

Concrete HLR data for services (e.g. CAMEL) can be considered as predefined data. Then the application model has to support predefined objects for each subscriber category. Customer categories might be grouped by selected services or tariffs.

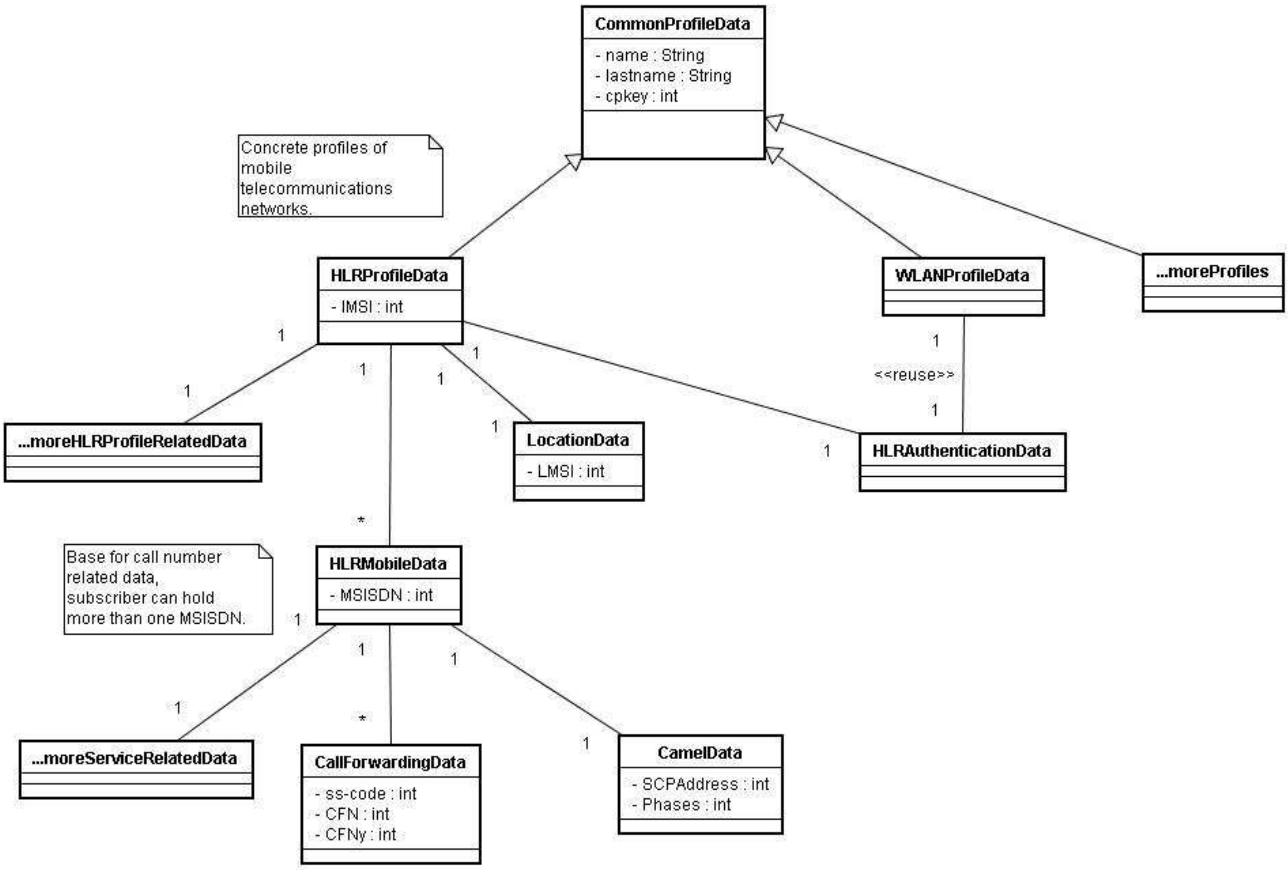

Fig. 6. High level class diagram of the profile data model

V. Conclusion

The introduced data model offers large flexibility for subscriber profiles of the different access networks ("plug-in" capability) of a single telecommunications provider. Reusability and data interoperability of the networks is guaranteed as well as component manipulation. The underlying object-oriented model supports the storage of multimedia and subscriber data. Also predefined profiles can be prepared. In comparison to the 3GPP General User Profile (GUP [5]) definition this approach integrates all data in a centralized database system. GUP uses redirect services or a proxy server architecture to access remote data from legacy systems. The centralized subscriber profile database is completely independent from the legacy systems, but supports legacy applications as well as new applications for complete subscriber profiles. Structural benefits of an object-oriented data model are the flexibility of service data integration and network data interoperability.

References

[1] López Aladros, R., Banet, F.-J., Rupp, S., Winter, E.: *Restructuring the Telecommunication Networks*, Working Conference HET-NETs'03, Ilkley, U.K. 2003.

[2] Banet, F.-J., Gärtner, A, Teßmar, G: *UMTS - Netztechnik, Dienstarchitektur, Evolution*, Hüthig Telekommunikation, Bonn 2004.

[3] Rupp, S., Siegmund, G., López Aladros, R., Banet, F.-J., *FLEXINET – A Network Service Architecture*, The Journal of The Communication Network, U.K. 2003.

[4] Russell, Craig : *Java Data Objects Version 1.0*, Sun Microsystems, Inc. All rights reserved. 901 San Antonio Rd., Palo Alto, California 94303 U.S.A., Java Community Process (JCP)

[5] Release 6 specifications:
3GPP TS 22.240 V6.5.0: *Service Requirement for the 3GPP Generic User Profile (GUP); Stage 1*
3GPP TS 23.240 V6.5.0: *3GPP Generic User Profile; Stage 1*
3GPP TS 23.241 V6.1.0: *3GPP Generic User Profile (GUP); Stage 2; Data Description Method (DDM)*
3GPP TS 29.240 V1.0.0: *3GPP Generic User Profile; Stage 3*
3GPP TS 24.241 V0.3.0: *3GPP Generic User Profile Common Objects; Stage 3*

[6] Salkintzis A.K, *The EAP-GPRS Protocol for Tight Integration of WLAN and 3G Cellular Networks*, VTC2003.

[7] Diehl A., Bleimann U., Fuhrmann W., Reynolds P., Rupp S., *Service Architecture for an Object-Oriented Next Generation Profile Register*, International Network Conference, Greece, 2005