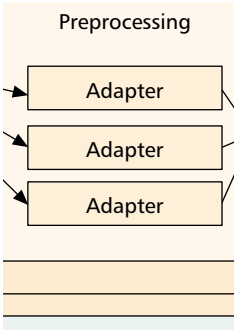
ADVANCED ADAPTABILITY AND PROFILE MANAGEMENT FRAMEWORK FOR THE SUPPORT OF FLEXIBLE MOBILE SERVICE PROVISION

NIKOS HOUSSOS, ATHANASSIA ALONISTIOTI, AND LAZAROS MERAKOS
UNIVERSITY OF ATHENS
EIMAN MOHYELDIN AND MARKUS DILLINGER, SIEMENS MOBILE RADIO NETWORKS
MICHAEL FAHRMAIR AND MAURICE SCHOENMAKERS, TECHNICAL UNIVERSITY OF MUNICH



The need is emerging for applying, in a systematic way, adaptability and reconfigurability concepts for service delivery in largely diverse contexts.

ABSTRACT

The long-term vision of beyond 3G wireless communications describes a mobile service provision environment dramatically different from that of today. Users are expected to raise their demands to a significantly higher level, toward the situation-aware provision of ubiquitous personalized multimedia services. From this perspective, the need is emerging to apply, in a systematic way, adaptability and reconfigurability concepts for service delivery in largely diverse contexts. Generic dynamically extensible adaptation mechanisms that can be employed in a wide variety of situations and are independent of the subject and criteria of adaptation is a significant step in this direction. Moreover, effective profile representation and management becomes an increasingly important issue. In the present article we introduce an advanced adaptability and profile management framework aiming to fulfill these requirements. The proposed system has been designed, implemented, and incorporated in a distributed middleware platform for next-generation mobile service provision.

INTRODUCTION

Service provision in second-generation (2G) mobile systems has inherited many characteristics of paradigms that were applied in traditional fixed telephony networks [1]: service logic is vertically placed in network nodes and tightly coupled with the underlying infrastructure; application development is performed by network operators or equipment vendors using specialized tools; access to services is exclusive to the subscribers of a particular operator and typically forms part of subscription-based offerings. Important negative effects of this approach include tedious and time-consuming (on the order of months or years) new service introduction as well as increased complexity of value-added applications development, resulting in a limited range of available services.

The above-mentioned restricted paradigms are gradually being enhanced by mobile data service provision models (e.g., i-mode, EZweb, and J-Sky in Japan) that are already being delivered over early deployments of 3G systems. The latter aim to provide a more flexible and dynamic environment [2], where a wide range of multimedia value-added applications, developed through partnerships among a variety of business players, will substantially enhance the end-user mobile service experience. Moreover, they are expected to be the first step toward the realization of environments commonly termed "beyond 3G" or "4G" [3], where a plethora of user-centric ubiquitous services, supported by computing capabilities embedded in all kinds of tangible objects and provided over heterogeneous reconfigurable network infrastructures, will significantly impact almost every aspect of our daily activities.

Reconfigurability and adaptability is appreciated as a key requirement for future mobile service provision management [4, 5]. In the heterogeneous environments envisioned for 4G, system and application functionality needs to be dynamically adapted to constantly changing contexts. This definitely requires that entities like system and service modules have the ability to be adapted. For example, common techniques for software adaptability have been developed based on component-based service development and/or interoperable service interface declaration and automatic discovery. However, one important aspect typically overlooked is the implementation of the adaptation logic itself. Notably, the criteria/parameters and adaptation decision logic are commonly static and hard-

1536-1284/03/$17.00 © 2003 IEEE

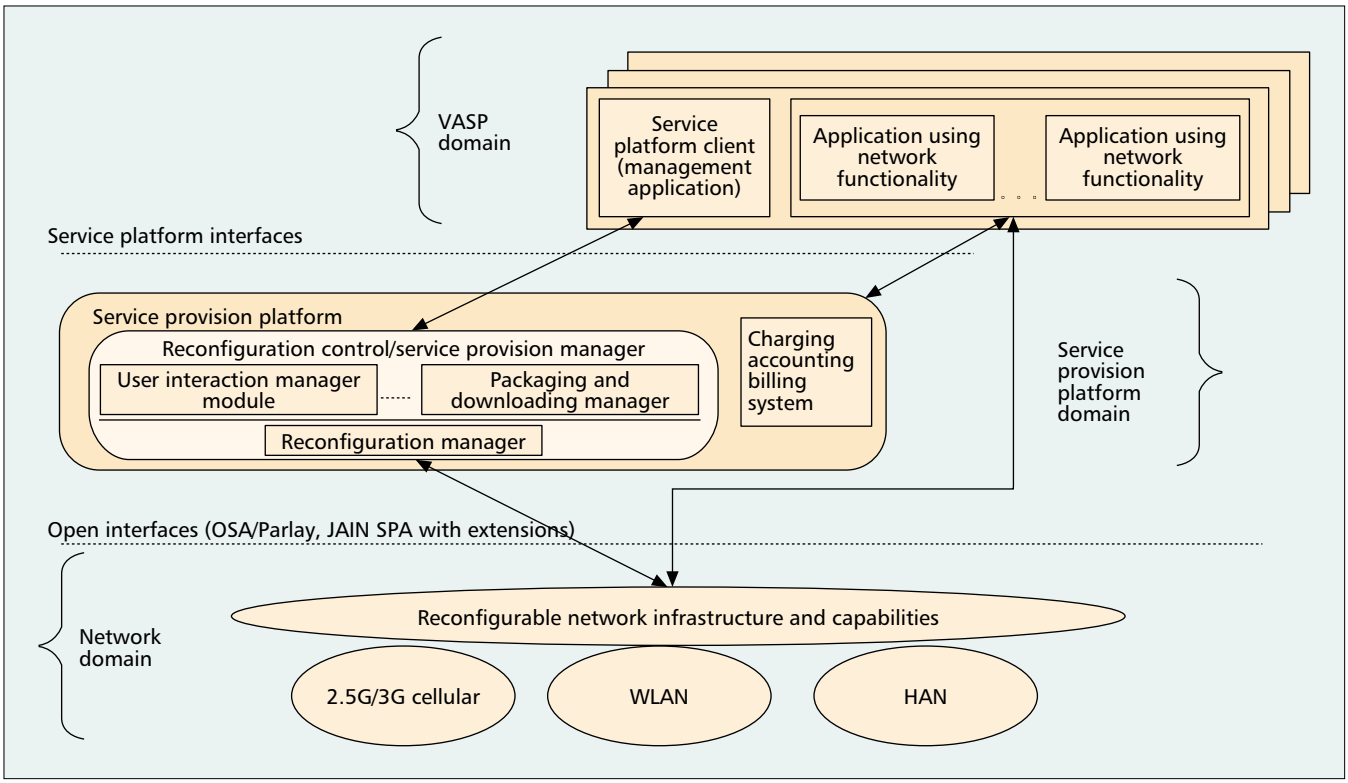


Figure 1. *Architecture for flexible service provision in 3G networks and beyond.*

coded in the adaptable entities. This appears not to be a problem in relatively static service provision environments like those encountered in 2–2.5G systems. In view of beyond 3G systems, though, this approach seems inadequate, since the circumstances in which an application/system function may be executed and the context parameters that may influence it will not be predictable a priori (e.g., at the time when the function is being developed). Furthermore, even the adaptation criteria and algorithms need to be adjusted over time according to the constantly changing surrounding environment. For instance, consider a 4G system that provides the user with the best possible connectivity for each service, given the available access technologies in the mobile terminal's area. Assume that a user travels in a rural area, where only Global System for Mobile Communications (GSM) and Universal Mobile Telecommunications System (UMTS) are the connectivity options, and intertechnology handover is based on relatively simplistic algorithms based on signal strength and required bandwidth by currently executed applications. However, as he/she arrives at the premises of a company where a larger number of wide-area as well as short-range access networks are available, the criteria and algorithms for access technology selection and vertical handover will be different; in this case more sophisticated policies and strategies can be activated, able to exploit the more detailed contextual information and increased network resources available, and reach optimal decisions by taking into account additional parameters related to factors such as interference, cost, security, and user status.

Generic reusable mechanisms that can cope with diverse adaptive profiles and adaptation algorithms present a solution to this problem. This article presents an adaptability and profile management framework and mechanism that aim to address these significant adaptation requirements of next-generation systems and services.

The rest of this article is structured as follows. We briefly present the architecture and functionality of a mobile service provision platform where the adaptability and profile management framework we propose has been integrated and validated. The adaptability and profile management concepts are introduced, and relevant requirements and issues of particular significance are presented. We then elaborate on the architecture, functionality, and implementation of the proposed framework. We briefly describe the use of the proposed framework in the aforementioned service provision platform. The last sections of this contribution are devoted to summary, conclusions, and acknowledgments.

A Framework for Mobile Service Provision

The proposed generic adaptation and profiling mechanism has been applied in the context of a middleware platform for the provision and management of value-added services over mobile networks in 3G and beyond [5]. The platform aims to address major issues regarding the delivery and management of downloadable services offered to users of next-generation mobile networks. These applications will mainly be provided by third-party software vendors, commonly

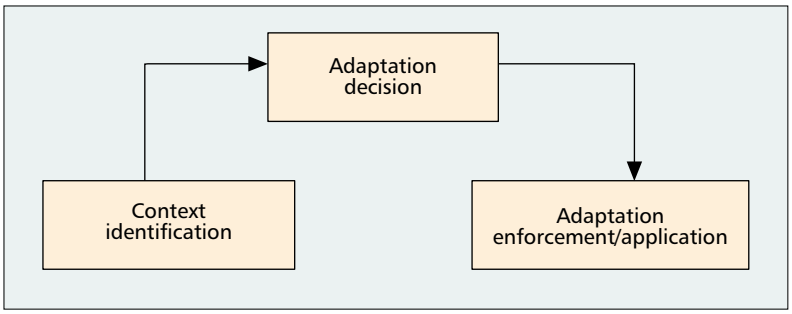


Figure 2. *A generic model of an adaptation procedure.*

termed *value-added service providers* (VASPs), instead of network operators and equipment vendors. The platform offers end users an intelligent context-aware mobile portal, where procedures like service discovery, downloading, and adaptation are fully tailored to terminal capabilities, user preferences, and network characteristics. Moreover, the platform enables VASPs to automatically deploy their applications over multiple network types (e.g., cellular, wireless LAN, home area networks), including facilities for reconfiguration of the underlying infrastructure through open application programming interfaces (APIs). The adaptation and profiling framework presented in this article is reused for all the above-mentioned customization actions.

The architecture of the platform is depicted in Fig. 1. The main components of this architecture are the following.

The *reconfiguration control and service provision manager* (RCSPM) is the central platform component in that it coordinates the entire service provision and management process. It includes modules that undertake online service deployment by VASPs, network reconfiguration, maintenance of service- and user-related data in suitable databases and repositories, as well as customized service discovery, downloading, and adaptation. The latter functions and, in general, all interactions with the user including the portal functions and control of the adaptation procedures (e.g., issuing requests to the adaptation module) are handled by an internal module of the RCSPM, the *user interaction manager module* (UIMM). Another critical module is the *reconfiguration manager* (RCM), which undertakes reconfigurability and adaptability management. The adaptation engine that lies at the core of the proposed adaptation mechanism is incorporated in the RCM. The UIMM acts as a client of this engine during the adaptation procedure.

The *charging, accounting, and billing* (CAB) [6] system is responsible for producing a single user bill for service access and apportioning the resulting revenue between the involved business players. This is accomplished by collecting metering data from appropriate network elements, like standard 3G core network components and enhanced metering-enabled IP routers.

The *end-user terminal* (EUT) [7] (not depicted in Fig. 1) resides in the user terminal and incorporates an execution environment for applications and functionality such as service downloading management, and graphical user interface (GUI) clients for service discovery and selection and capturing of event notifications. The EUT, depending on the equipment available at the terminal (e.g., GPS devices, sensors), is able to communicate to the RCSPM information useful to the adaptation mechanism, like terminal capabilities, and user status and location.

Introducing Adaptability and Profile Management in Mobile Service Provision

Adaptability Management

Adaptability is commonly defined as the ease with which an entity can be dynamically modified according to its context. Dynamic modification (adaptation) can be considered a transition to another state of the entity and could include an alteration in the entity's behavior. Context refers to any information that can be used to characterize the situation of the entity[1] [8].

Considering the applicability of the above definitions in the field of mobile service provision, one could identify several types of entities that can be subject to adaptation:

- End-user services. Potential adaptation actions for this entity type are the addition/removal/switching of service components and modifications in the physical distribution of service elements.
- Middleware and management applications and functions, like the platform presented earlier, that support the creation and provision of mobile services. For instance, functions like service discovery and downloading can be highly customized based on the current environment.
- Protocols at various layers of the networking protocol stacks. Examples include dynamic selection of error correction schemes depending on the link layer technology employed, or the dynamic activation of transport protocols optimized for communication over wireless links.
- Control and management plane mechanisms like horizontal and vertical handover, dynamic selection of access technology, and network instrumentation.
- Content provided to the mobile user (e.g., adjustment of transmission parameters of audio/video streams, transformation of the presentation format of the data included in a Web page).

It is obvious from the above that the concept of adaptation to context encompasses a large variety of mechanisms that can be applied to disparate entities of mobile systems and services. Naturally, the adaptation of entities as diverse as an error correction scheme, a handover algorithm, and the contents of a Web page would be performed in quite dissimilar ways. However, the process of adapting an entity can in general be modeled in three distinct phases, as depicted in Fig. 2.

1) Context data retrieval/monitoring, which corresponds essentially to collecting the appropriate information from the current environment and typically involves:

[1] *Many definitions of the notion of context exist in the literature. The one we have adopted was introduced in [8].*

- Discovery of available context data sources.
- Collection/monitoring of context data from the appropriate sources, which may include methods like periodic polling or subscription to specific events of interest. This task can be supported by techniques such as caching and replication of context data, which enable better performance, availability, and reliability.

2) Determination of the most suitable (in the present situation) entity state. This procedure usually consists of a number of subtasks:

- Obtaining relevant context information (e.g., metadata about the requirements/features of each alternative) and optionally preprocessing of this information (e.g., format conversion, filtering, consolidation) so that it is suitable for input to the decision functions.
- Identification of the decision algorithm/criteria. A relevant subtask is determination of the types of context information that are useful in the situation at hand.
- Performing the actual decision function, which typically involves matching and processing of profiles describing the adaptable entity and the current context. During this task only the context information useful in the current situation is taken into consideration. The outcome of the function may prescribe the transition to another state/configuration of the entity.

3) Application/activation of the selected transition, which comprises:

- Retrieval of the adaptation results.
- Employment of entity-specific mechanisms for activating (putting into effect) the best alternative.

The design of adaptive systems based on this model can greatly benefit from two observations that have attracted little attention in existing systems. First, it is useful that the implementations of the different adaptation phases are decoupled from each other. Important aspects of this decoupling are the following:

- Algorithms and strategies for making adaptation decisions are not aware of how the context is being collected and vice versa.
- The mechanisms for putting into effect adaptation decisions are independent of how decisions are made and vice versa.
- The mechanisms for putting into effect adaptation decisions are independent of what context parameters are used in adaptation and how this data is collected.

This decoupling leads to modularity of adaptation code; individual module implementations pertaining to a certain phase can be modified/replaced (either at compile time or runtime) without affecting the components of the other phases. The benefits of such a decoupling (however, in a more limited sense) have also been recognized in the literature [8–10].

The second observation concerns the fact that a number of adaptation functions can be totally unaware of the *type* of entity that is subject to adaptation. Entity type independence for a function implementation essentially means that:

- There is no compile time awareness of any profile entity or context profile parameters processed by this function.
- Data about any specific algorithms employed for the accomplishment of this particular function is not known at compile time. Thus, the algorithm logic is not hard-coded in the function's implementation.

Thus, entity type independence requires that all entity-type-aware data and logic be loaded at runtime. The obvious benefit of this approach is that entity-independent function implementations can be reused for adapting various types of entities in diverse environments. This way, entities can be more easily enhanced with adaptation logic. Furthermore, a lack of such a feature would hamper scalability, since the support of a variety of adaptation data and processing logic would be achievable only by increasing the size and complexity of the corresponding system. Last, and perhaps more important, entity type independence facilitates dynamic modification of the adaptation factors (e.g., context parameters that affect adaptation, algorithms for adaptation decision making) for an entity during the entity's lifetime. For instance, different criteria for adaptation may apply in different access technologies and surrounding environments, since certain parameters (e.g., in terms of security, resource constraints) that are critical in a specific context are less significant or can even be ignored in another situation (as also described earlier).

Naturally, a number of constituent parts of the overall adaptation process (e.g., identification of the context parameters that are relevant for a particular adaptation instance, enforcement of an adaptation decision) are inherently dependent on the type of adaptation subject (whether it is an end-user service, link-layer protocol, or Web page). However, certain adaptation subtasks can be developed to be entirely agnostic to the semantics of the adaptation operation that is underway as well as the type of adaptable entity. In particular, making adaptation decisions, including profile processing and employment of the appropriate algorithms, can be handled (as shown later) by a generic component whose functionality is appropriately called by entity-specific adaptation components through an open interface.

Developing a generic adaptation mechanism suitable for highly dynamic environments like mobile networks and services in 3G and beyond is an excessively demanding task. A summary of the fundamental requirements would be:

- Clear demarcation of the logic implementing the different phases of the entire adaptation process.
- Support for arbitrary complexity. The adaptation mechanism should support elements, like profiles and algorithms, of arbitrary complexity.
- Independence from particular types of profiles and algorithms. The algorithms used for profile matching should be loaded at runtime and thus should not be a static part of the adaptation mechanism. That is, any logic pertaining to specific types of profiles and algorithms should not be hard-coded in the adaptation system.
- Interoperability and portability. Ideally, the adaptation system should not only handle disparate data and algorithms, but also enable its seamless integration in a variety of environ-

Entity type independence requires that all entity-type-aware data and logic be loaded at runtime. The obvious benefit of this approach is that entity-independent function implementations can be reused for adapting various types of entities in diverse environments.

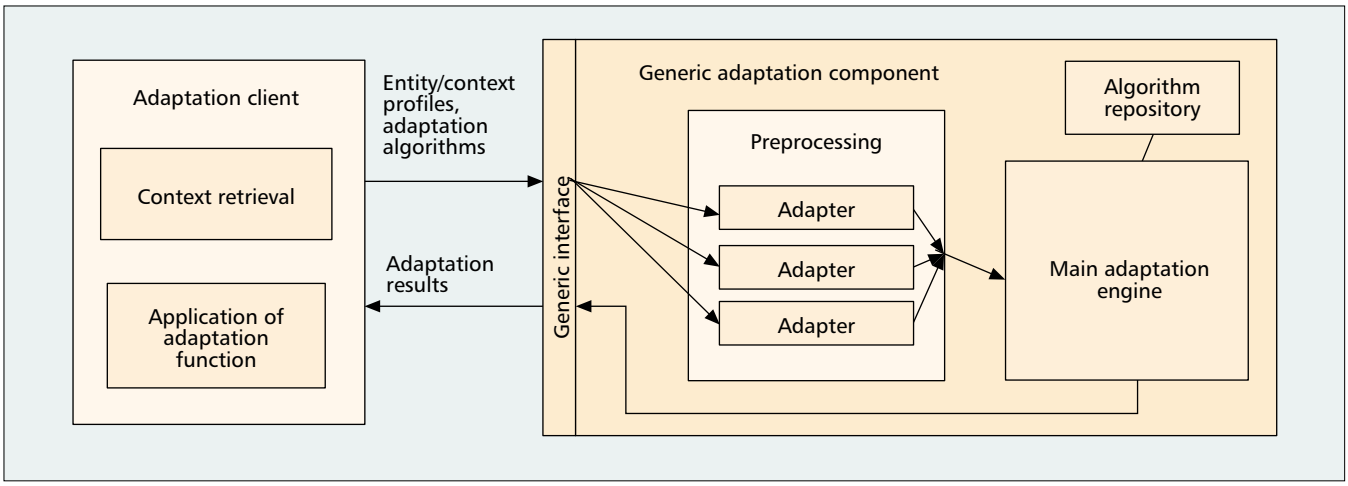


Figure 3. *Architecture of the proposed adaptation system.*

ments. System modularity and decoupling from context are vital to the accomplishment of this goal.

Profile Representation and Management

The implementation of adaptation functionality requires the processing of a significant amount of meta-data, describing the adaptable entities as well as the context elements. This information can be expressed in terms of profiles. Entity profiles should contain data that indicates the association/dependence of the entity variants to particular context parameters. For example, an adaptable application profile should clearly specify the requirements of the individual application versions/configurations in terms of execution environment capabilities, bandwidth, etc.

A desired feature in profile management is the adaptivity/dynamic extensibility of profiles. In many occasions context data in its entirety contains more information than the sum of the different context parameters. For example, the fact that terminal battery is low and the user had scheduled a download with high priority can be used to derive a preference of the user for switching to a faster but more expensive access network. Such dynamic preferences could have immediate impact on the infrastructure, triggering its reconfiguration and improving many aspects of quality of service provision. A variety of ways can be used for preferences derivation. Examples include neural networks, rule-based expert systems, and statistical evaluation.

Another aspect relates to the unified modeling of context and entity profiles that could facilitate the dynamic introduction of adaptation functionality, since it enables the specification of adaptation behavior to operate on a homogeneous model for both context and entity changes.

Additionally, the concept of meta-attributes as described in [11] for communication profiles can be applied to a context model that is suitable for specifying adaptation behavior for service provisioning systems. Meta-attributes describe generic attributes of context information like accuracy, change, and request frequency or predictability, and allow for several optimizations to provide efficient access and management of context information in wide-area distributed, wireless, or low-bandwidth environments.

In summary, important issues/requirements in the design of a profile management system include the following.

Profile representation (semantics/syntax). Semantically rich profiles, containing data about arbitrarily complex entity attributes and relationships should be supported. Profile syntax should be characterized by clarity and, to the extent possible, simplicity so that it enables easy development and interoperability of profile implementations. Furthermore, a unified view/representation of profiles of disparate entities facilitates profile handling.

Extensibility/adaptivity/interoperability. Since it is hard to predict all the parameters/factors that could make up a beyond 3G service provision context, as well as the constraints and additional information deriving from their combination and interdependencies, there should be no restrictions on the types of profiles supported. The runtime introduction/derivation of new profile attributes as well as new profile types (e.g., new types of context elements) should be supported.

Efficiency. Profile management, inherently a complex task, could lead to significant consumption of resources (e.g., network capacity, memory, persistent storage). Therefore, the careful distribution of profile data to appropriate locations and selection of collections mechanisms (e.g., push vs. pull), and the employment of sophisticated optimization schemes (e.g., caching, compression) are required to avoid negative consequences on efficiency.

Generic Adaptation and Profile Management Framework

The generic adaptation system presented in this section concentrates on the adaptation decision phase of the overall adaptation procedure. It provides a generic adaptation decision engine that is entirely decoupled from the other two phases (context retrieval and activation) and can

thus be reused unaltered in a variety of adaptation operations in different situations. The engine, which has been prototypically implemented in Java, is totally agnostic of specific profile parameters or algorithms. All data needed for the adaptation decisions is dynamically loaded; thus, compile time knowledge is not required. This has been achieved through the appropriate use of object-oriented design patterns and reflection mechanisms.

The rest of this section describes the architecture of the proposed framework, elaborates on profile representation and implementation issues, and summarizes the most significant features of our approach.

Architecture and Functionality

The architecture of the proposed generic adaptation mechanism is depicted in Fig. 3.

The generic adaptation component exposes its functionality to the adaptation clients through an open interface, shown in Fig. 4. Through the interface, clients feed into the component the profiles of the adaptable entity and the corresponding context, and optionally also the adaptation algorithms. In return, they get the entity profile, suitably adapted according to the current context, and also may be transformed into another format (depending on the adaptation algorithm). The latter transformation may optionally be performed by the client instead of the adaptation component.

Internally, the system is divided into two parts: preprocessing and the main adaptation engine.

The preprocessing module undertakes the transformation (if necessary) of the entity/context profiles to the internal common format employed by the adaptation engine. The preprocessor identifies the current format of the profiles and either forwards them directly to the main adaptation engine (if they are in the right format) or dispatches them to the appropriate module for the required transformation (the design of the preprocessor makes use of the adapter pattern). Adapters for various formats can be seamlessly plugged into the system, enhancing its functionality without affecting the rest of the components.

The adaptation engine returns to the client the entity profile adjusted on the basis of the provided context information, according to an algorithm either provided by the client or contained in a local repository as default for the specific entity and context types. The engine has the additional capability to load algorithm implementations from remote network locations. The loading is performed according to appropriate identification information (e.g., combination of class names and codebase URLs).

Profile Representation and Implementation Issues

The adaptation engine accepts as input two profiles, the *adaptor* and *adaptee*. The adaptor is a profile encapsulating context information used for adaptation. For instance, in 3G service provision the adaptor could represent the capabilities of a terminal, the preferences of a user, and the characteristics of a network. The adaptee denotes the entity that needs to be customized according to the adaptor data (e.g., a protocol or an application). A single adaptee may also describe entity collections and/or hierarchies with arbitrary size, structure, and semantics. Essentially, the adaptee profile has the form of a tree. Each node in the tree represents a customizable unit identified by name and type that could be part of a larger structure. Moreover, every adaptee node contains profile(s) of adaptor type, which indicates the capabilities/preferences/requirements of the corresponding unit pertaining to the specific adaptation criteria expressed by an adaptor. A special degenerated case would be the adaptee representing an atomic nondecomposable entity and thus taking the form of a tree with a single node. Each adaptee profile instance aggregates an object that encapsulates the decision/matching algorithm. This is due to the fact that the matching algorithm may vary not only among various types of attributes (e.g., String, ScreenSize) but also between different instances of the same attribute type.

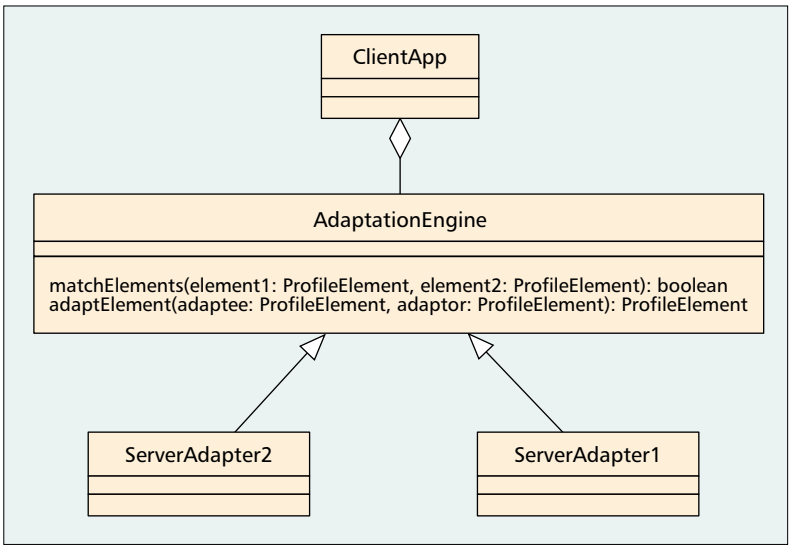


Figure 4. *High-level class diagram and interface of the adaptation engine.*

The class diagram of Fig. 5 depicts the common internal representation of profiles as Java classes in our implementation (the presented class interfaces are not complete for readability reasons). Notably, the adaptee and adaptor profiles are instances of the same class (*Profile*). Profile objects are instances of profile elements, and consist of single profile attributes and subprofiles. This is an application of the Composite design pattern [12], which has been chosen due to the fact that a profile may consist of plain attributes as well as nested profiles, and there is a common set of operations that should be included in the public interfaces of both plain attributes and compound profiles. The main operation that can be invoked on a *ProfileElement* is the adapt() method, which encapsulates the algorithm for adapting a ProfileElement object according to another ProfileElement object of the same type.

Alternatively, the adaptation algorithm may be encapsulated in a separate object that imple-

The proposed mechanism is designed to be generic, so that it is able to operate in diverse 4G service provision environments without any modifications to its code or recompilation.

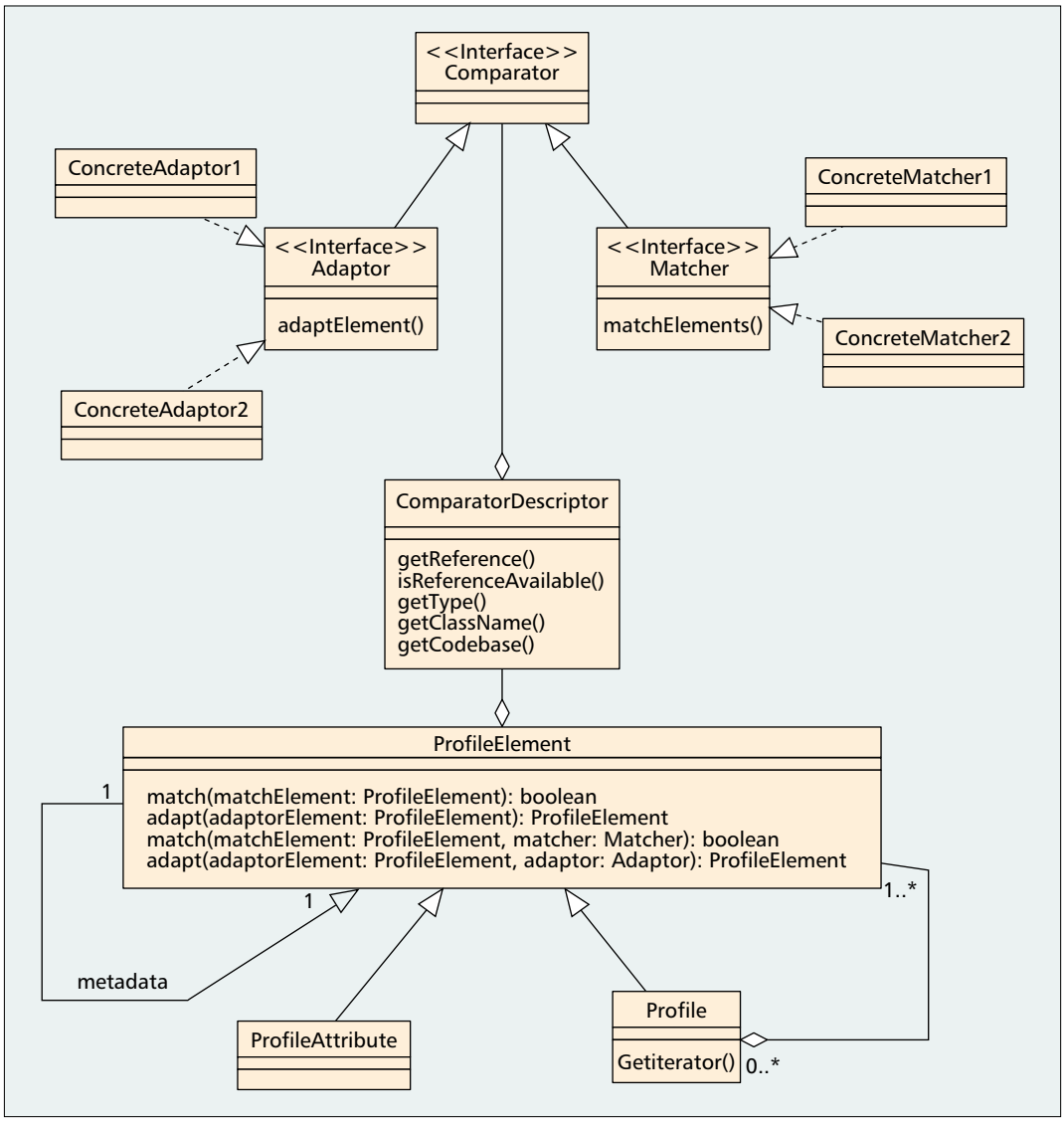


Figure 5. *Class diagram of profile representation objects.*

ments the *Comparator* interface (and typically also the Adaptor interface). Such an object is aggregated in a ProfileElement instance and can be dynamically loaded at runtime (this is an application of the Strategy design pattern [12]). This way, new adaptation algorithms for specific attributes (anything from simple pattern matching to fuzzy logic and neural networks) can be introduced without the need to modify or recompile the code of the ProfileElement implementations that represent those attributes. Thus, third-parties (e.g., VASPs) are able to tailor the context-aware decision behavior of the adaptation system to their needs (e.g., to the requirements of a particular service), without modifying the system itself. Information concerning the algorithm implementations (e.g., the network address from which they can be retrieved) is contained in *ComparatorDescriptor* objects.

A degenerated case of adaptation occurs when the adaptor and adaptee are profiles of the same type, and the adaptation result required is only a Boolean indication of whether the two profiles match; the algorithm is then encapsulated in either the match() method of the ProfileElement or an object that implements the *Matcher* interface. The match() methods and Matcher interface are not strictly necessary, but are included in the design for programming convenience, since the special case in which they are used occurs frequently for a typical adaptation client.

Each ProfileElement contains a single reserved attribute called *metadata*. This is a generic container (it is itself a ProfileElement) that integrates all the available metadata related to a specific context profile. It is worth stressing that this information does not pertain to the entity represented by the profile, but to the quality of the profile data; it implements the meta-attributes concept in [11]. Apparently, the set of applicable types of metadata depends on the profile type, for which different types of metadata elements make sense (e.g., attributes like estimated accuracy and various parameters that concern when, where, and by whom this profile has been retrieved). However, we have identified certain parameters (e.g., time of profile element retrieval, administrative domain from which the

profile element was acquired) that apply to a wide range of context data and have been employed for all the context information types we used in our prototype.

Evaluation of the Proposed Solution

The proposed mechanism is designed to be generic so that it is able to operate in diverse 4G service provision environments without any modifications to its code or recompilation.

Features of particular importance include:

- The system is able to handle profile data structures as well as adaptation/matching algorithms of arbitrary semantics and complexity.
- Entity profiles and adaptation algorithms are dynamically loaded at runtime. Thus, context data types and algorithms unknown or not implemented at the time of development of the adaptation system can be seamlessly used. Moreover, important adaptation factors like the kind of context data used or the algorithms employed can dynamically change during the lifetime of an entity instance (e.g., service, protocol). This is possible because the adaptation system code is completely independent of the particular profiles and algorithms used for every adaptation operation. A relevant important implication is that these adaptation algorithms can be dynamically loaded by third parties that are the creators/owners of the adaptable entity (e.g., VASPs that have developed an adaptable end-user application).
- Arbitrary mechanisms for context retrieval can be applied, since they are completely transparent to the adaptation engine.
- The format of the adaptation output[2] can be tailored to the client's needs without any modification to the adaptation system, just by loading an appropriate Comparator object that produces output in the desired form.
- The system itself can be dynamically extended to support new representations of context data through runtime plugging of new adapters for appropriate translation of content formats.
- The adaptation engine is stateless; thus, every request is served independent of any previous ones by the same client. This contributes to the autonomy of the adaptation component and allows its distribution and replication in a straightforward manner. This characteristic enables our system to scale well when the number of requests increases.

Applying Generic Adaptation for Flexible Mobile Service Provision

To better illustrate the concepts and mechanisms presented earlier, in this section we elaborate on how our framework is used in the context of the service provision platform presented above for service discovery and delivery to the end user. First, we provide an overview of how these operations are accomplished, followed by a presentation of the corresponding interaction sequence between platform components.

Overview

In the middleware platform described earlier, the proposed adaptation mechanism is applied for procedures like customization of listings of available services as well as determining per request the components of an application that should be packaged by the RCSPM and downloaded by the user terminal.

For instance, when a mobile user wishes to view which services are accessible in its current context (let's assume that context in this case comprises terminal capabilities, user preferences, and network characteristics), the service provision system should retrieve the list of available services (from an application database/repository) and filter it according to context. We assume that each service may come in multiple versions, and that service- and version-specific contextual requirements, together with information on where the corresponding adapting algorithms may be retrieved from, are obtainable from the service database. In this case, the adaptor would consist of the terminal, user preferences, and network profiles, while the adaptee would comprise a list of services, with each service typically consisting of a list of available versions. Each service version can also contain optional components that are included or not in the downloaded service, again depending on the service provision context. This service information is arranged in a tree, with every node containing identification information for the service/version it represents, the corresponding terminal and network requirements, as well as the supported user preference attributes.

We assume that each service may come in multiple versions and that service- and version-specific contextual requirements, together with information on where the corresponding adapting algorithms may be retrieved from, are obtainable from the service database.

The output of the adaptation component is a customized version of the adaptee. In the above example, the adaptation result would have the form of a tree structure containing the list of matching services (e.g., as a matching service could be considered one with at least one version that is compatible with the current context), with each service list entry, including a list of compatible versions for the specific service. Moreover, each service version entry would contain the optional components that should be incorporated in the service in the current context. This adaptation result in our service provision platform is used for presenting customized service listing to mobile users, as well as for the on-demand packaging (including dynamic inclusion of optional components in service package) and downloading of services to mobile clients.

The UIMM is the module responsible for coordinating the above-mentioned procedures and issuing requests to the adaptation component. Moreover, it undertakes the collection of context data during a user session with the platform. However, different types of profiles come from different origins and in different formats. Service and user profiles are obtained by querying the corresponding databases maintained by the platform operator, terminal capabilities are sent to the RCSPM by the mobile terminal, while network characteristics can be retrieved from the underlying infrastructure through appropriate open APIs. Conversion to the profile representation scheme presented earlier is performed through appropriate adapters that can be dynamically added to the system. The UIMM also performs caching of the profiles so that data that has been retrieved from persistent storage can be efficiently reused until the end of

[2] *By adaptation output we mean the adaptee entity profile after the completion of the adaptation operation.*

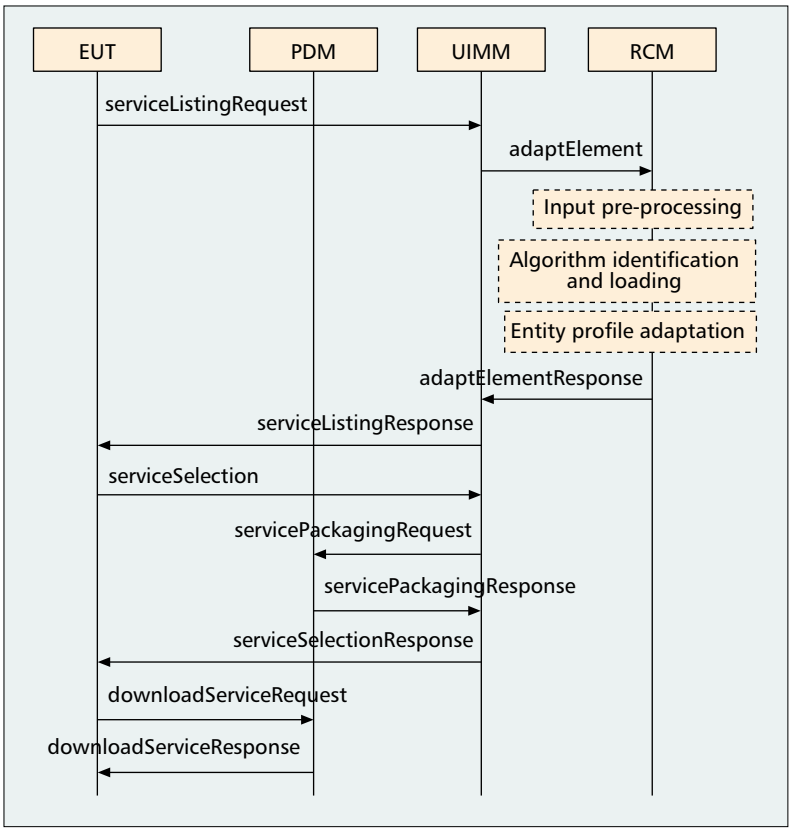


Figure 6. *Interaction sequence for service discovery, selection, and downloading.*

the user session for processing subsequent user requests. The adaptation functions are handled by the generic adaptation mechanism, as described earlier, that is incorporated in the RCM of the RCSPM, which accepts generic matching queries and returns the results to the UIMM that appropriately processes them. The dynamic packaging of services, which is based on the adaptation results, is performed by the packaging and downloading module (PDM).

Example Interaction Sequence

In this section we describe a scenario of service adaptation in the proposed platform, which is useful for illustrating the application of the concepts described above. This is presented in Fig. 6, which depicts the interactions between the user terminal and the service provision platform, required for service discovery, selection, adaptation, and downloading. These interactions are always preceded by successful initiation of a user session with the RCSPM, consisting of security-related operations like mutual authentication.

The aforementioned functions are performed through the following phases:

1) The user issues a request for a service listing, with criteria like service category and keywords. Contextual information, including terminal capabilities and also optionally network characteristics and ambient environment data, are sent to the UIMM as parameters of the request.

2) The UIMM retrieves from the service database the profiles of applications that match the user query and tailors the service listing to environmental conditions. To achieve that, it collects all the required service profile and context data and forwards it to the adaptation module in the RCM, which identifies and returns (*adaptElementResponse*) a list of the profiles of the services that are suitable for delivery in the current context. These profiles include information about available service versions and the optimal configuration (e.g., proper values for initialization parameters, inclusion or not of optional components, physical distribution of service elements) for each version. This task is accomplished in three phases:

- Preprocessing and format transformations (if necessary) of the input profiles, as described earlier.
- Identification and loading of the decision algorithm. The algorithm implementation is loaded from the local repository (if it is available there) or from a remote network location, whose address is obtained through the corresponding ComparatorDescriptor object.
- The execution of the algorithm, which results in deciding on the best adaptation option and producing the adapted entity profile.

Notably, the latter two functions above are recursively invoked for each one of the nested profiles contained in the entity profile.

3) The customized listing is returned to the user.

4) The user selects a service from the list.

5) The UIMM triggers the PDM to perform packaging of the adapted service. It is worth noting that the detailed service profile adaptation results related to appropriate application configurations, which were produced in phase 2, are used as an input parameter of the servicePackagingRequest to PDM. After completion of this task, the UIMM returns to the EUT the URL of the corresponding customized package.

6) The user downloads and executes the customized service.

7) Steps 1–6 can be repeated as many times as the user desires to discover and access other services.

Summary and Conclusions

The significance of adaptation in next-generation mobile systems and services is widely recognized. Adaptation capabilities form a principal enabler of ubiquitous seamless service provision over highly diverse underlying infrastructures. Sophisticated, flexible adaptation mechanisms are required for the realization this vision. The present article introduces a generic adaptability and profile management framework that exploits object-oriented design patterns and reflection features to support increased runtime flexibility, generality, and reusability of adaptation functionality. The proposed framework has been prototypically implemented, and incorporated and validated within a distributed mobile service provision platform.

Acknowledgments

The authors would like to acknowledge the valuable contribution of Mr. Spyros Pantazis in the design and implementation of an early version of the adaptation component prototype.

REFERENCES

[1] A. Lazar, "Programming Telecommunication Networks," *IEEE Network*, Oct. 1997.

[2] UMTS Forum Report no. 9, "The UMTS Third Generation Market — Structuring the Service Revenues Opportunities," http://www.umts-forum.org/

[3] J. Pereira, "Fourth Generation — Beyond the Hype, A New Paradigm," *Proc. IEE 3G Mobile Commun. Tech.*, London, U.K., Mar. 2001

[4] Markus Dillinger *et al.*, Eds., *Software Defined Radio: Architectures, Systems and Functions*, Wiley, 2003.

[5] A. Alonistioti and N. Houssos, "The Need for Network Reconfigurability," in [4].

[6] M. Koutsopoulou, A. Kaloxylos, and A. Alonistioti, "Charging, Accounting and Billing as a Sophisticated and Reconfigurable Discrete Service for Next Generation Mobile Networks," *Fall VTC 2002*, Vancouver, Canada, Sept. 2002.

[7] O. Fouial. K. A. Fadel, and I. Demeure, "Adaptive Service Provision in Mobile Computing Environments," *Proc. 4th IEEE Int'l. Conf. Mobile Wireless Commun. Networks (MWCN 2002)*, Stockholm, Sweden, Sept. 9–11, 2002.

[8] A. K. Dey, "Providing Architectural Support for Building Context-Aware Applications," Ph.D. thesis, College of Comp., Georgia Inst. of Tech., Dec. 2000. http://www.cc.gatech.edu/fce/ctk/pubs/dey-thesis.pdf

[9] B. D. Noble, M. Satyanarayanan, "Experience with Adaptive Mobile Applications in Odyssey," *Mobile Networks and Applications*, vol. 4, 1999, pp. 245–54.

[10] B. Badrinath *et al.*., "A Conceptual Framework for Network and Client Adaptation," *Mobile Networks and Applications*, vol. 5, 2000, pp. 221–31.

[11] E. Mohyeldin, M. Fahrmair, and C. Salzmann, "Communication Profiles," in [4].

[12] E. Gamma *et al.*, *Design Patterns: Elements of Reusable Object Oriented Software*, Addison Wesley Longman, 1995.

BIOGRAPHIES

NIKOS HOUSSOS (nhoussos@di.uoa.gr) received a B.Sc. degree from the Department of Informatics at the University of Athens in 1998 and an M.Sc. (with distinction) in telematics (Communications and Software) from the Department of Electronic and Electrical Engineering, University of Surrey, United Kingdom, in 1999. Since 1999 he is pursuing a Ph.D. at the Department of Informatics and Telecommunications, University of Athens. He is also a staff member of the Communication Networks Laboratory of the University of Athens. He is involved in projects MOBIVAS (including workpackage leadership), ANWIRE, and POLOS of the European Union IST framework. His main research interests are middleware platforms and business models for beyond 3G mobile service provision, service adaptability and network reconfigurability management.

NANCY ALONISTIOTI (nancy@di.uoa.gr) has a B.Sc. degree and a Ph.D. degree in informatics and telecommunications from the University of Athens. She worked for seven years at the Institute of Informatics and Telecommunications of NCSR "Demokritos" in the areas of protocol and service design and testing, mobile systems (UMTS), open architectures, CORBA, intelligent networks, and software defined radio systems and networks. She specializes in the formal specification and testing of communication protocol and services, design of object oriented SDL platforms, mobile applications, reconfigurability, adaptable services, design of test cases for conformance testing using TTCN, design of IN services and CORBA-based architectures and services. She has participated in several national and European projects (CTS, SS#7, ACTS RAINBOW, EURESCOM, IST-ANWIRE etc.), and is technical manager of the IST-MOBIVAS project. She had been working as an expert at the National Telecommunications Commission, and she is currently project manager, senior researcher in the CNL (University of Athens), and a member of the project management team of the Greek Academic Network — GUNET. Her current research includes mobile systems, software reconfigurable radio systems and networks, adaptability, service management and download, and design of open architectures and platforms.

LAZAROS MERAKOS (merakos@di.uoa.gr) received a Diploma in electrical and mechanical engineering from the National Technical University of Athens, Greece, in 1978, and M.S. and Ph.D. degrees in electrical engineering from the State University of New York, Buffalo, in 1981 and 1984, respectively. From 1983 to 1986 he was on the faculty of Electrical Engineering and Computer Science at the University of Connecticut, Storrs. From 1986 to 1994 he was on the faculty of the Electrical and Computer Engineering Department at Northeastern University, Boston, Massachusetts. During the period 1993–1994 he served as director of the Communications and Digital Processing Research Center at Northeastern University. During the summers of 1990 and 1991, he was a visiting scientist at the IBM T. J. Watson Research Center, Yorktown Heights, New York. In 1994 he joined the faculty of the University of Athens, Greece, where he is presently a professor in the Department of Informatics and Telecommunications, and director of the Communication Networks Laboratory (UoA-CNL) and the Networks Operations and Management Center. His research interests are in the design and performance analysis of broadband networks, and wireless/mobile communication systems and services. He has authored more than 150 papers in the above areas. Since 1995, he is leading the research activities of UoA-CNL in the area of mobile communications, in the framework of the Advanced Communication Technologies and Services (ACTS) and Information Society Technologies (IST) programs funded by the European Union (projects RAINBOW, Magic WAND, WINE, MOBIVAS, POLOS). He is chairman of the board of the Greek Universities Network , the Greek Schools Network, and a member of the board of the Greek Research Network. In 1994 he received the Guanella Award for the Best Paper presented at the International Zurich Seminar on Mobile Communications.

EIMAN MOHYELDIN (eiman.mohyeldin@siemens.com) received her B.Sc. (Honors) in electronics/telecommunications and control from the University of Gezira, Sudan, in 1993, and her M.Sc. degree from Technical University of Munich, Germany, in 2000. In 2001 she joined the Mobile Network Division at Siemens and developed software for a layer 1 EDGE signal processing unit. Later she has joined the system engineering division where she was responsible for SDR design of network management and driving concepts for reconfigurable systems .She has published several technical papers mainly in the field of mobile communication systems, software-defined radio, and radio resource management.

MARKUS DILLINGER [M'00] (markus.dillinger@siemens.com) received his Diplom-Ing. degree in telecommunications in 1990 from the University of Kaiserslautern, Germany. In 1991 he joined the Mobile Network Division at Siemens and was developing call processing software for the GSM base stations. Later, he joined the system engineering division where he was responsible for the evaluation software for the Siemens Channel Sounder measurement equipment. In addition, he was responsible for the definition of the technical requirements and specification on the GSM base station line interfaces. From 1995 on, he worked on the definition of the third mobile radio generation in the European research project, FRAMES, and in 1999 he was appointed technical manager of FRAMES. Since January 2000 he has been the project leader of the European research project TRUST and the follow-up project SCOUT for software radio. He has published many articles in the field of WCDMA and software radio.

MICHAEL FAHRMAIR (fahrmair@informatik.tu-muenchen.de) received his Diplom-Inf. degree in computer science in 1998 from the University of Technology, Germany. Currently he is completing his Ph.D. in computer science at Munich University of Technology, Germany, while working as a researcher under the chairmanship of software and systems engineering of Prof. Dr. Manfred Broy. His main research field is mobile and ubiquitous computing.

MAURICE SCHOENMAKERS (schoenma@in.tum.de) received his Diplom-Inf. degree in computer science in 1994 from the Munich University of Technology, Germany. After working several years in industry as a consultant, he is currently completing his Ph.D. in computer science at Munich University of Technology, Germany, under the chairmanship of software and systems engineering of Prof. Dr. Manfred Broy. His main research field is the specification of service-oriented architectures and mobile systems.

Adaptation capabilities form a principal enabler of ubiquitous seamless service provision over highly diverse underlying infrastructures. Sophisticated, flexible adaptation mechanisms are required for the realization this vision.