Integration Issues in Communication Environments

Dmytro Zhovtobryukh

Department of Mathematical Information Technology, University of Jyväskylä,
40014 Jyväskylä, Finland

Abstract. The integration of diverse communication systems is currently one of the primary evolutionary trends in telecommunication networks. The ultimate goals of the integration are ubiquitous and seamless service and communication provisioning across the variety of the platforms and unprecedented mobility freedom. Terminal-based, network-based and service-based network integration approaches are briefly examined in this paper. However, the paper mainly focuses on the service provisioning aspect of network integration. Transition from vertical to horizontal services will facilitate a network interoperability solution on the service level. Current research efforts are focused on the exhaustive service characterization that is necessary in order to facilitate the development of adaptive service applications and to split the service provisioning process into independent design and provisioning efforts.

1 Modern Perspective

Today's telecommunications world features a number of communication technologies and is subjected to serious technological diversity. On the other hand, the current technologies are, to a certain extent, functionally narrow. Moreover, since most of them are approaching their maturity, it is extremely difficult, if not impossible, to either introduce significant technological enhancements to them or cease using them in favor of the other technologies. Modern communication standards can be roughly split onto fixed and mobile technologies. The fixed sector is much more mature and well developed. It is almost entirely presented by the Internet, the nearly global computer network system. The Internet relies on interconnected packet-switched wireline networks, which allow provisioning of rich data and multimedia services at high speeds. This, along with the robust network vehicle and global accessibility, makes the Internet a peculiar communication standard. On the opposite side are the mobile communication technologies that primarily utilize circuit-switched voice-oriented wireless networks and, most importantly, establish advanced support for user mobility, which is missed in the Internet. However, wireless technologies constantly suffer from low data rates, and therefore lack support for data and multimedia.

Recently, new technological challenges have been presented to communications with the emergence of the "next generation networks" concept. Most of them are predictable and strive to preserve the most valuable features of the second-generation networks and the Internet. Next generation networks will be characterized by superior quality in wireless communication at the highest speeds and with advanced reliability.

Zhovtobryukh D. (2004).
Integration Issues in Communication Environments.
In *Proceedings of the 1st International Workshop on Ubiquitous Computing*, pages 28-37
DOI: 10.5220/0002664800280037

Rich data and multimedia services, along with traditional voice capabilities, will be supported. Future networks must facilitate unprecedented mobility freedom and anytime access from anywhere. Service applications must express intelligent behavior. The list of the features to be added is endless. However, the most-mentioned challenges are not new. The research areas of the Mobile Internet, 3G networks, and Ubiquitous Computing have been facing them for a long time.

The Internet could be a perfect foundation for next generation communications because of its global coverage, robust network vehicle and high speed. Furthermore, it has been the base of the information society for years, and the majority of services have been already developed within it. These services require mere adjustments to meet the demands of tomorrow. Unfortunately, the fixed nature of the Internet makes it unsuitable to face the challenges of mobility support and ubiquitous access. Mobile computing poses new requirements to communication systems in addition to further complicating the fundamental challenges inherent in distributed systems. Frequent handoffs and disconnected operation force a system to become ever more dynamic and adaptive. Ubiquitous access creates further challenges to the terminal equipment and to the network system in order to handle dynamic, integrated and personalized access to the available services.

Modern communications apparently are moving towards a global, integrated computing environment. The idea of such a convergence is to combine the existing communication standards into one single communication system, while preserving their advantages and overcoming their drawbacks. There are two major ways to achieve that. The revolutionary approach consists of the design of a totally novel technological standard that is effectively compliant to the above demands. It may imply a purely novel technological basis, software and hardware architectures, although the best achievements of the modern technologies could be reused. Its first reflection the approach found within the Ubiquitous Computing paradigm.

An alternative way is rather extensive and can be called an evolutionary one. It focuses on the integration of the existing communication standards within a single system. This concept currently seems to be more reasonable since it does not require a complete reconstruction of the existing systems. Instead, it entails the technological reinforcement of today’s communication platforms in order to create an integrated global communication environment. However, it appears substantially complicated but may make contemporary communication systems interoperable. Terminals, network architectures, services and even users may be highly diverse in the various systems. Significant modifications must be made to them to introduce a unified system platform that powers their interoperability. Whether such integration is tight or loose depends on many factors, such as how flexible the system is to modifications, and which degree of modifications is affordable and reasonable for network operators.

2 Ubiquitous Computing

The concept of *Ubiquitous computing* (Ubicomp) was initially proposed in the early 1990s [1], [2] and particularly established to denote a new paradigm of distributed computing. Being user-centric rather than the traditionally service-centric, Ubicomp focuses on the demands of a user encompassed by a computing environment. The environment is no longer seen as a separate computing facility; rather it is embedded into the physical environment so that the user lives in it naturally. The concept of *invisibility* arises from this point. As a matter of fact, truly appropriate and advanced technology must disappear into the background, allowing the user to use it unconsciously but deliberately. Conversely, the modern personal computer brings itself to the foreground and, as a consequence, a user not only makes a conscious decision to use it, he/she must learn in advance how to use it. The fork and knife that people use while having meals can be a good example of invisible technology. Another important aspect of Ubicomp is *immersion*. Since the computing environment is embedded in the physical neighborhood and is effectively invisible to the user, one becomes immersed in a new type of reality that combines the physical surroundings with the computing system. In contrast to virtual reality environments, the real world in an immersive system is not replaced by an imaginary one but is just altered appropriately. Invisibility and immersion are the properties that make computing really natural and user-friendly.

In Ubicomp a computing environment highly populated with computing devices embraces the entire physical surroundings. The user can get access to computing facilities from anywhere within the environment. The computing devices dwelling within the environment are networked through wireless and/or wired media on a certain technological basis. They represent conventional tools for acquiring access to and manipulation of the services that are available through the environment. Those devices are as simple as possible to learn and to use. They are envisioned to replace such standard communication tools as pens, paper, work desks, cord phones, etc., and to automatically store, process, and manage information via the specific applications. Applications would be seen not as an interface to the user but as a specific functionality for accomplishing service-related tasks, similarly to personalized user agents. *Personalization* is a subject of great importance since, in the Ubicomp paradigm, terminals are not necessarily personalized for every single user, but applications most likely are. The applications should be even able to migrate between terminals, tracking the user’s movements and performing the tasks on the move (the “follow-me” principle) [4]. Finally, the applications are adaptive. They are capable of adjusting their behavior according to the changes occurring through user’s demands, communication conditions and the computing environment. The majority of the tasks can be automated in ubiquitous computing. Due to *adaptability*, applications can autonomously manage personal information, schedule appointments, find other users within the environment, and so on. *Context-awareness* is a capability of the applications to take into account various environmental data and their changes for consequent adaptation of their behavior. Such data is widely referred as *context* [5].

So far, Ubicomp addresses an interesting contemporary viewpoint of distributed computing. In addition to being a more natural paradigm for communications from the user’s point of view, it also addresses and resolves in a graceful and efficient manner some crucial challenges of modern computing [6] but poses new ones [3]. Despite the fact that Ubicomp environments are still a long way from being deployed widely, some of the solutions proposed within this approach can be effectively adopted now to settle the same challenges in current communication environments, such as ubiquitous access, mobility, and adaptive applications.

3 Integration of Communication Environments

Three principal integration approaches can be considered to enhance interoperability among the originally non-interoperable standards. Terminal-based, network-based and service-based approaches address different subsets of interoperability and offer particular solutions for system integration, which are to be seen as complementary, rather than exclusive or competitive. A combination of all three approaches would be the most comprehensive solution for the network integration problem.

3.1 Terminal-Based Approach

The terminal-based approach essentially features a functional convergence of the end terminals so that they possess a number of necessary capabilities to enable ubiquitous access throughout heterogeneous communication environments. As a result, the underlying network standards should not be strictly interoperable because the terminal - be it laptop computer, pocket computer, or mobile phone - handles the network access functions itself, depending on the network accessed. However, this does not settle the problem of cross-network service provisioning and service integration. Services originally residing in certain network systems are not available if this network is currently inaccessible, i.e., out of the terminal's range. But even if multiple network systems can be accessed simultaneously, the integration of two services designed for different systems is yet impossible. Appropriate modifications have to be made to the applications in order to integrate those services. Moreover, although the networks themselves remain unaltered within this approach, certain modifications to the provider's service network would be needed to handle the integrated billing and subscription. Another crucial factor is terminal complexity. If two or three diverse communication systems are accessed, the complexity might be still acceptable. But with a growing number of systems, the terminal complexity will become a serious flaw of the approach. Therefore, the described solution can be effectively applied only to integrate a small number of platforms, e.g., in GPRS-WLAN internetworking [7].

3.2 Network-Based Approach

The main idea of the network-based approach is to rebuild communication networks appropriately so they become interoperable. In essence, this means bringing the common network vehicle inside all of the communication networks that are subject to interoperability. The Internet layer 3 network protocol (IP) is most likely to become such a common network vehicle. This indeed appears to be a reasonable solution since IP is the robust protocol, widely used, and the majority of applications have been already developed for IP networks. Unfortunately, IP suffers the drawback of being oriented toward fixed network access. Therefore, substantial research efforts are focused on the design of a mobility-driven version of the protocol, widely referred to as Mobile IP [8], [9]. However, the IP addressing of mobile network nodes seems to be a complex problem that is further complicated by the necessity of support for cross-network mobility. So far the approach is challenging but not yet full-fledged.

Should it be successful, this approach would seal the network-based interoperability challenge and allow the direct interconnection of the IP-based networks into a global system. However, a disadvantage of the approach is the possible redesign of the services from former non-IP networks. Instead, this will enable service portability and service integration among all IP systems. Such a solution would expand the understanding of the Internet and force the implementation of the Mobile Internet.

3.3 Service-Based Approach

The principal idea of this approach is to separate the phase of service design from the service application load and run inside a particular communication network. First of all, such separation would create a base for service portability across the diversity of communication environments. Secondly, due to a generic service design, it would substantially simplify service integration. However, this does not seem to be possible without the introduction of a unified programming model for service application development and its implementation on top of a number of communication networks. The model also requires special mechanisms that decide how to create and/or adjust applications with respect to specific systems and on the basis of the given programming model. These mechanisms comprise what is called *middleware* [10], [11]. *Reflective* middleware [12] is an intelligent interface for a dynamic application adaptation/customization in compliance with the specifics of a concrete communication system, service properties and context changes in the computing environment. The approach is widely used but is still far from its ideal state. Currently, there are several completed middleware standards that are even interoperable with one another. However, universal applications capable of operation on top of the multiple middleware platforms are still subject to development.

This approach is primarily in the research scope of this paper. The main ideas powering this approach are best described within the concept of *Pervasive computing* [13] and are as follows:

Abstract view of services. In order to make services portable, they should be presented in a rather abstract way, e.g., only general functionality of the service is specified. All of the network, terminal and user-specific aspects of service presentation would not be determined within the service itself, but within the application. The application presents certain functionality for the user to manipulate with the service in the way that the user prefers and in the form his/her terminal allows. Thus, the application concretizes the generic service to a specific functionality and presentation. Service portability across different environments is enabled via accounting on environmental specifics in every concrete application.

Service integration and discovery. The services populating the computing environment could be combined on demand to create some form of integrated functionality for accomplishing sophisticated tasks. If such functionality can be discovered within the services available through the environment, the application can call on the appropriate service and compile the functionality of multiple services.

Adaptation of applications. Since applications represent services customized to the form suitable for a concrete network, terminal and user, and since those factors tend to dynamically change, the applications need to be adaptable to possible changes in the environment. Appropriate reaction of the application to any changes occurred should preserve the quality and convenience of the service being provided. Adaptability itself means only that the application can change its behavior and knows how to perform, e.g., it may apply different strategies for achieving its goals, or can reconfigure its own structure appropriately.

Context-awareness. Adaptive applications capable of reacting to environmental changes necessitate the use of mechanisms for tracking and interpreting those changes. Such information is referred to as *context* and requires treatment to be present in the form appropriate for applications. Environment monitoring, context acquisition, and context interpretation are generic mechanisms of context-aware systems. An important issue for both adaptability and context-awareness is the place within the system where these features are built. Different approaches try to balance the distribution of the adaptation mechanisms between the application itself and the system (middleware). Giving total control of adaptation procedures to the application makes the program too large and complicated, whereas the system, which manages the adaptation, is less reliable and requires significant modifications.

However, passing the responsibility of shaping and adapting the service to the system allows researchers to separate the service provisioning into the distinct stages of service creation and service delivery. Service creation here corresponds to the process of designing generic service functionality that consequently can be provided in various ways for different environments and contexts. The service delivery phase is responsible for creating an application based on the generic service or set of services, but for a concrete environment. Here one can distinguish between the application load phase and the application run phase. At load phase, the application is being constructed from the set of generic elements [14]. Generic service is one of these elements. The other elements are related to various environmental characteristics, e.g., the terminal form-factor, medium type, quality requirements, contexts of interest, etc. At this stage, the application is enforced with certain strategies that define the range of its possible behaviors. Obviously, at the load phase, the application can be defined as portable across multiple heterogeneous environments. In this case the application should contain necessary strategies for other environments as well. Meanwhile, the application at run phase is just operating in a way determined at its construction. Whenever essential context changes occur, the application may change its behavior by choosing another strategy from those available at the construction. The handoff to another environment is also an essential change. Having detected it, the portable application adapts to a new environment by picking up an appropriate strategy.

Speaking about service-based interoperability, it is necessary to assume that some basic mechanisms for roaming among heterogeneous networks would be established and the networks themselves would be somehow physically interconnected.

4 Service Portability Framework

Service portability is the ability of the service to be reused across multiple communication systems. Current understanding of the service in communications is rather static and flat. Once created, the service can no longer be altered during its lifecycle. For the development of portable services, two main perspectives exist. The first is to build portability directly into service. In this case, service is portable, but is limited to a set of environments and cannot be altered to map other emerging possibilities. Furthermore, service definition becomes tremendously huge with the increase of the number of supported environments.

Another approach is to free the service from any details concerning the environment of its use. So, the portability is built into a particular service application instead. Whenever the need for porting the service arises, the appropriate application is constructed or an existing one is modified accordingly. So far, a particular service application can be created for either a portable or a local use. For each supported environment application has a set of operational strategies for appropriate behavior adaptation. Such a service portability framework is a core aspect of the service-based integration approach.

4.1 Scenarios

In order to illustrate how the service portability framework operates, let us consider a few of the most general operation scenarios. The home network is understood as the network where the service originally resides. A foreign network is a network to which the service is supposed to be ported.

The first scenario corresponds to the case in which a user at foreign network requests a service (see Fig.1.a). This case can be named as *static*, since no efforts for run-time adjustments of the service application are needed. In the most general case applications are constructed and launched on demand. So far, the user's request for a service must result in an appropriate service application launch in the foreign network. Assuming a generic service design, the application can be issued in a foreign environment as so it is intended for a local use in this environment. Thus, a static service portability scenario illustrates the importance of a generic service design.

Figure 1.b illustrates a *dynamic* service portability scenario. A user in home network requests the service and then moves to another network. The application is constructed and launched while the user is connected to the home network. But when the user with her terminal makes a handoff to another network, the application is perhaps no longer suitable for service delivery. One alternative way to resolve the situation is to terminate the current application and to issue a new one in the foreign environment. The main flaw of this option is the service session discontinuity. This may be unacceptable to the user, who expects seamless service delivery.

To preserve access transparency and service quality, the application should be adapted to a new environment instead of being re-launched. In this case, a service session persists during the handoff and the user is relieved from possible service terminations. Being handed over to another environment the portable application senses the environment and selects an appropriate strategy or reconfigures itself.

The dynamic service portability scenario addresses the situation of Single Terminal and Multiple Environments (STME) with the terminal mobility involved, whereas the static scenario corresponds to Single Terminal and Single Environment (STSE) without mobility. However, there is a case of multiple terminals (MTSE and MTME) with the user mobility involved.

MTSE case is basically irrelevant to the integration problem. Therefore, only MTME case is considered (see Fig.1.c). The scenario makes principal difference from STME from the adaptation point of view. The application must not only adapt to

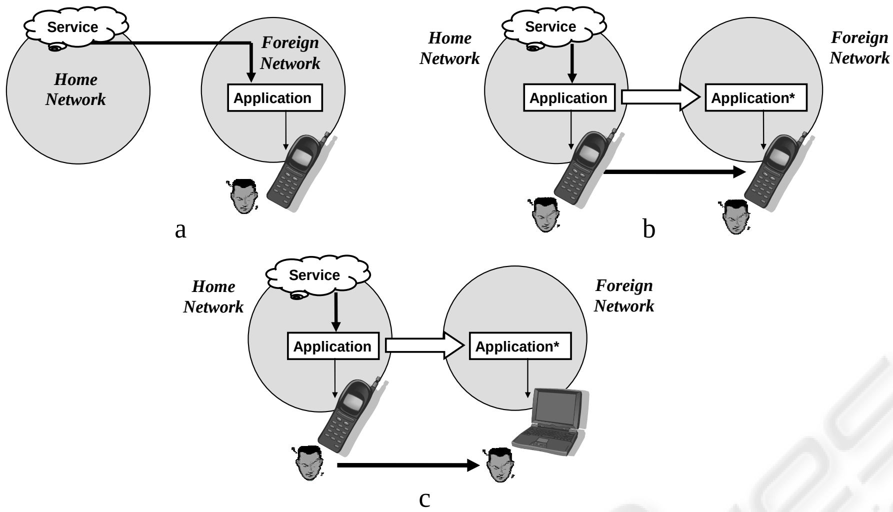

Fig. 1. Service portability scenarios: a. static service portability; b. dynamic service portability; c. "follow-me" dynamic service portability

another environment, but also find a terminal to which the user switched and adapt to that terminal. So, the application gets two execution contexts: network and terminal.

The "follow-me" principle for tracking user mobility in pervasive systems has been previously used for development of Teleporting applications at Olivetti Research Lab [4]. This experience can create a foundation for portable "follow-me" applications.

Portability of service applications across diverse networks becomes possible because of applications' adaptability. Generic service design benefits in the existence of unique service instance that needs no alteration to be reused among heterogeneous environments. The burden of service adjustment is given under control of applications. This can also facilitate service integration inside an application. The last but not least issue to be addressed is adaptation frameworks for portable applications.

4.2 Adaptation of Portable Applications

The separation between service creation and service delivery necessitates considering a service not a black box but the object that has an internal structure and properties. Despite the service obtains only the abstract description on the stage of creation, it is concretized to the form of application on the delivery stage. The service is processed several times to adsorb the specific properties that correspond to the concrete context. The initial abstract form of the service would be seen as a rough core that needs an appropriate shaping. In face of different contexts, such as terminal form factor, network bandwidth, etc., the same abstract service may result in a variety of specific service applications. Such a vision is more complex than current service provisioning frameworks, but is more flexible. Once created, the service can be modified dynamically with respect to the momentary conditions found in the environment.

The services that users are used to receive are vertical services. A traditional vertical service is what a user gets at her end terminal after the service has been processed and shaped appropriately. More specifically, the sequence of modifications the abstract service undergoes to obtain its final form is the vertical service. Metaphorically, at the service load stage the initial abstract service accumulates implementation-specific properties like a snowball rolling down a hill. Every time the service acquires one more layer of specifics it becomes less general and more concretized. In reality, these layers of specifics – let us call them abstraction layers – should be precisely defined. Otherwise, adaptation strategies of the service applications cannot be designed. Abstraction layers can be user-, software-, hardware- and network-related. They should be arranged in a certain type of vertical infrastructure in particular order to reflect the sequence of modifications the service undergoes on its path from its abstract to the final form. In such service architecture service modifications propagate down to the bottom layer, while the constraints established by each layer - to the top. Basically, every abstraction layer comprises multiple variable characteristics that describe the service-related functionality of the layer. The generic service gets concretized as its application is adapted to the requirements of every abstraction layer. The layer-specific functionality features a set of variable characteristics that comprise a horizontal service. A horizontal service is an entire functional capacity of a single abstraction layer. Every single value of its characteristics is a service primitive, a minimal service-related construction unit. The service primitives are precisely the bricks of which a vertical service is constructed. When environmental conditions are changing, particular service primitives on certain abstraction layers are reselected to match the changes occurred.

Adaptation strategies built in the service application represent such particular sets of service primitives that reflect concrete context situations in the environment. For example, when a user with a terminal moves to another network, environmental conditions immediately change, e.g.; different connectivity, traffic classes, etc. These context parameters correspond to concrete service primitives on the concrete abstraction layers (network, transport). When the application realizes that the context changed, it selects such primitives that match the observed conditions. The set of newly selected primitives identifies the new strategy, which the application will use to operate adequately in the new environment. In other words, the application uses the horizontal services that are available in the new environment to construct a valid vertical service in this environment.

However, to implement the described adaptation framework, it is necessary to introduce new types of middleware services that are capable to derive service primitives and provide them to the applications. The first step is the design of a service abstraction model with an appropriate layering. Such an attempt has been made in [16]. After it is developed and validated, every horizontal layer must be thoroughly examined for service-related mechanisms and their properties. This would create a base for exhaustive service characterization. The service characterization implies determining exhaustive set of service primitives in each horizontal service, relationships between the primitives and their natural combinations for different types of vertical services. If exhaustive service characterization can be achieved, then the described reflective middleware can be developed, which would force the development of adaptable and portable service applications.

5 Conclusions and Future Work

The rapid proliferation of telecommunications revealed the solid trend towards serious technological divergence. Convergence is highly desirable instead to introduce a global communication environment. Revolutionary approaches propose complete redesign of existing systems. As a compromise, evolutionary approach implies complex integration to be solved on all of the layers of network architecture.

The service portability is the core concept of the service-based integration. Portable services can be reused across communication environments. To develop portable services, avoidance of specific service design is a must. Separation of service creation and service delivery allows for unique but generic service instance in interconnected environment. Such service design also facilitates service integration.

Essential place in the service portability framework occupy adaptable applications. As the service implementations they perform actual service porting across the environments. Passing this function to the application complements to the generic service design. Applications customize services to various environmental contexts. They are launched on demand and can adapt at run-time, without service interruption.

Future work will focus on practical issues of design of service reference model with appropriate horizontal layering and service characterization on top of the model.

References

1. Weiser, M.: The Computer of the 21st Century. Scientific American (1991)
2. Weiser, M.: Some Computer Science Issues in Ubiquitous Computing. CACM (1993)
3. Demers, A. J.: Research Issues in Ubiquitous Computing. CACM (1993)
4. Bennett, F., Richardson, T., and Harter, A.: Teleporting – Making Applications Mobile. Proceedings of IEEE Workshop on Mobile Computing Systems and Applications (1994)
5. Floch, J., Hallsteinsen, S., Lie, A., and Myrhaug, H. I.: A Reference Model for Context-Aware Mobile Services (2002)
6. Satyanarayanan, M.: Fundamental Challenges in Mobile Computing (1996)
7. Davoli, M.: WLAN as a Complement to GPRS and 3G Services. White Paper, Ericsson (2002)
8. IETF Mobile IP Working Group: IP Mobility Support. RFC 2002 (1996)
9. Perkins, C.: Mobile IP. IEEE Communications, Vol.35, No.5. (1997)
10. Raatikainen, K.: Nomadic Computing Research Space (2001)
11. Raatikainen, K.: Functionality Needed in Middleware for Future Mobile Computing Platforms (2001)
12. Keeney, J., Cahill, V.: Chisel: A Policy-Driven, Context-Aware, Dynamic Adaptation Framework (2002)
13. Banavar, G., Beck, J., Gluzberg, E., Munson, J., Sussman, J., and Zukowski, D.: Challenges: An Application Model for Pervasive Computing. Proceedings of Mobicom 2000 (2000)
14. Raatikainen, K., Hohl, F., Latvakoski, J., Lindholm, T., Tarkoma, S.: Generic Service Elements for Adaptive Applications (2001)
15. Chen, G., and Kotz, D.: A Survey of Context-Aware Mobile Computing Research. Dartmouth Computer Science Technical Report TR2000-381 (2000)
16. Zhovtobryukh, D., and Kohvakko, N.: Service Reference Model for Modern Communications. To appear at Fourth International Network Conference INC2004 (2004)