A VERSATILE NETWORK INDEPENDENT SERVER ARCHITECTURE FOR MULTIMEDIA INFORMATION AND COMMMUNICATION SERVICES

Wolfgang Kellerer
Munich University of Technology, Institute of Communication Networks
Arcisstr. 21, 80333 Munich, Germany. Email: kellerer@ei.tum.de

Abstract The success of new service architectures will substantially depend on how they cope with existing technologies. Information and communication technology is characterized by a multitude of heterogeneous transmission platforms on which a multitude of different systems provide services to the users. The latter dispose of various terminals and varying network access points. Despite of one global communication standard the heterogeneity is even supposed to increase in the near future. For flexible service provisioning and to support various heterogeneous technologies, new concepts have to be worked out addressing especially the rising customer requirements and providing network access to new market entrants.

This paper presents a new component based server architecture that realizes convergence of heterogeneous systems on a service level, abstracting from the networks. The control of information and communication services is strictly separated from the control of communication bindings in the networks. Network adaptor units enable the connection to any network type including packet oriented networks and broadcast networks. The server architecture is targeted to high-level services rather than basic services that are already provided in the networks. Especially, the combination of previously separated services allows a variety of new service offers. The structure of the service architecture components according to a session concept in access, service, and communication session supports an easy service control and management. The signaling between the components with simple messages bases on the principles of the IETF SIP concept. Service and user mobility features based on a directory server concept and a service discovery mechanism for adaptive selection of resources support flexible service provisioning. The functionality of the server architecture is illustrated by a detailed example.

1. INTRODUCTION

The market for information and communication services is presently characterized by quick changes. Permanent technical improvements and the rising demand for electronic communication in general lead to a competition situation, where providers come up with new services in shorter and shorter life cycles. In addition, the progressing liberalization in telecommunications fastens these changes in service provisioning. New market entrants show up and have to differentiate themselves from the incumbent operators in prices and (even more important) by offering unique services. Since most parts of networks are still under control of former monopolists, new approaches are needed to open networks to provide new services in a flexibe and quick way to the users.

Important improvements have also been made in the underlying network techniques, which contribute to the rapid changes in the services market. New and advanced transmission techniques (photonics, wireless) and improvements in data processing (processors, memory) provide increasing bandwidth and lead in general to enhanced capabilities of communication networks. Increasing service offers and decreasing prices also raise the demand of users for more quality and for more supplementary features in their information and communication services. Competing providers have to address this demand: Existing services are enhanced with new features (e.g. supplementary services in PSTN and mobile networks by the Intelligent Network), new services are being developed (e.g. GPRS for wireless data communication) or new services are being created by the combination of previously separated services. This has led to a rising variety of generally available services and service systems (e.g. ISDN telephony/fax, wireless telephony, SMS, GPRS, WAP, WWW) as well as personalized services (e.g. unified messaging, one number services).

We can also observe an increasing variety in the control of communication networks. Unlike the attempts to build a homogeneous global network infrastructure (as with B-ISDN/ATM in the 90ies), currently even more and differentiated network technologies and protocols come up (e.g. H.323, HSCSD, GPRS). In the same way it cannot be predicted, whether the next generation system for wireless communication (UMTS) in fact is able to integrate all future services. In addition, packet oriented IP-based networks move in the focus of interest, even though they lack in providing reliable quality for telecommunication services. Beneath that, digital broadcast networks gain importance. In this way not the attempt to *one* global infrastructure but to an infrastructure consisting of multiple heterogeneous parts can be observed.

Similar to the networks, the systems enabling information and communication services – the service architectures – are also very heterogeneous. Network specific architectures like POTS exist beneath service integrating architectures like ISDN. Application specific architectures like T.120 or DSM-CC complement the variety.

The user requests services that are easy to handle, allow personalization, can be provided in short time, and fullfil all their complex demands. Customized services should be accessible independently of specific networks and terminals.

One approach that supports the necessary convergence is provided by the IP technology. With the IP protocol, different transmission networks converge in one superior transmission protocol on the network layer. However, IP based services are terminal based. This means that all service software has to be integrated in the terminals and kept consistent. The IP approach is not sufficient for the above stated demands for this reason. In addition, IP based networks lack in providing features that are indispensable for telecommunication services. Guaranteed quality of service, security mechanisms, regulatory aspects, and billing concepts are missing.

To find a more suitable solution, we have to consider on the one hand the requirements and on the other hand the heterogeneity of the standards. A feasible approach would be to support the convergence of the different networks and systems on a *service level* superior to all the existing architectures. Service architectures that realize this concept would bring together existing networks and systems to provide best usage for the customers. According to the antagonistic principles of the location of network intelligence for service control (intelligent networks vs. intelligent terminals) we can call this "Intelligence on top of the networks".

Along with this more network centric approach this paper describes a component based server architecture. The control of information and communication services is strictly separated from the control of communication relationships within the networks. Adaptation units realize the connection to any heterogeneous network. The server architecture is targeted to high-level services rather than basic services that are already provided in the networks. Especially the combination of previously separated services allows a variety of new service offers. Beneath well-known telecommunication services and their enhancements by supplementary services especially broadcast services and asynchronous services are focused for service combinations. The structuring of the service architecture components according to a session concept in access, service, and communication sessions reduces complexity for service control and management.

There are other approaches that as well address the problem of infrastructure independent service provisioning. [1] describes an architecture for service provisioning for hybrid services based on programmable networks. Service control components are distributed among network nodes and terminals. This programmable network approach does not fit with our requirements, since it affects infrastructure and user equipment. Other approaches are based on migration issues towards high-level architectures like TINA and propose the integration of heterogeneous networks therein e.g. [2, 3], but only consider legacy telecommunication networks.

This paper is structured as follows. First we give an overview over the evolving techniques for multimedia service architectures and point out the requirements for future architectures. Section 3 describes our approach that deals with the mentioned requirements by introducing a novel network independent server architecture. The key concepts of this architecture are presented in Section 4, whereas Section 5 describes some personalization and mobility features. Finally Section 6 illustrates the functionality with a service example. A summary and an outlook on further work are given in the conclusion.

2. MULTIMEDIA SERVICE ARCHITECTURES

To convey further understanding of the basic principles of our architecture and to elaborate the motivation, we will first discuss the characteristics of existing service architectures regarding their suitability according to our requirements (see also [4]).

Telecommunication Service Architectures. Telecommunication service architectures are evolving from many service specific networks (e.g. POTS, X.25) to multi-service systems. Narrowband ISDN has been the first step towards an integrated services network. The service control of ISDN is tightly integrated within the switching protocols and therefore very network specific. Broadband ISDN is based on the same principles, however.

The Intelligent Network (IN) enhances the public ISDN (PSTN or mobile networks) with a centralized control of supplementary services, which is separated from the switching system. Neither user terminals nor the user network interface need to be touched for the provisioning of supplementary services. But the development of services is restricted to a limited call model resulting from the PSTN.

IP Based Service Architectures. The internet considered as a service architecture provides a platform for flexible service provisioning by the common open interface of the IP protocol. However services e.g. email, ftp, WWW, are running on the end systems. That means the appropriate service software has to be present in all involved terminals and has to be integrated there for any new service. Looking beyond simple information services, we will see in addition that the IP architecture is lacking important features indispensable for telecommunication services (e.g. QoS, security, and billing mechanisms are missing).

Even architectures for IP telephony, like H.323 of the ITU-T [5] or SIP of the IETF [6], are not fulfilling all requirements for flexible multimedia service provisioning yet. H.323 is in accordance to ISDN quite protocol oriented and allows only a very limited implementation of non-standardized services, since it is limited to the ISDN call model. Based on simple textual messages the SIP IP telephony is more suited for open service development. In addition SIP is transaction oriented similarly to HTTP, i.e. call states are not stored longer than the call establishment phase. Therefore services are not restricted to an unflexible call model. But on the other hand no influence from within the network on the services is possible after their initiation.

Furthermore both IP telephony architectures outline the need for a central network intelligence, located e.g. in a network server, for reliable service control. H.323 proposes the use of a Gatekeeper whereas SIP uses SIP servers (Proxy Server, Redirect Server).

TINA. The internet concept provides a network layer platform for all entities interworking for a service. High-level service architectures like TINA [7] instead are based on a complex middleware for the interconnection of components on application layer. In TINA the middleware (the TINA consortium proposes CORBA) carries the signaling between all components of the service and network architecture. Hence a great amount of flexibility for the provision of new services can be provided. However the middleware has to be present in every terminal and in each switching node. This of course restricts a wide-spread usage. Moreover TINA is focused on an ATM/B-ISDN infrastructure. Nevertheless the service architecture concepts very well support decomposition into separated functional areas (access session, service session, communication session). The TINA session concept based on a network centric service session manager allows a much more generic control of services than the telephony call models.

To summarize, we can see that none of the existing architectures as they are is qualified for an overall architecture to realize the convergence of networks on the service level. Especially the correlation of network characteristics within the service control is too tight. Other concepts are not suited for a wide spread usage yet since they affect the software in the terminals. A centralized approach seems to be advantageous for the control of complex services since it provides the necessary overview independent from the current status of a certain terminal.

Towards APIs. In addition to the service architectures discussed above, a number of approaches are emerging that define programming interfaces (API) for services. The IETF standardizes for example an access point from the internet to IN Service Control Points (SCP): PINT [8]. The PARLAY Group defines an open interface to allow third party providers to control services over a PARLAY API [9]. Both approaches illustrate the trend towards open interfaces for previously closed networks and are very well suited as APIs for superior network independent architectures.

Terminals and Customer Requirements. As we have seen internet services for example require powerful PCs as universal internet terminals, on which the service software is running. Existing terminals for telecommunication services, e.g. telephones or mobiles, instead are application specific. Also in future, specific terminals will exist and even be preferred since they are more adapted to certain use cases, e.g. PDAs. To provide access to personalized services we will have to improve on the service server side.

New Requirements. As we have worked out, an effective way to cope with the existing heterogeneity of terminals, networks, and service architectures will be an overlay service provisioning of all infrastructures together. This would address the communication demand and reflect the various customer access points. A service architecture is supposed to support the following features to provide this kind of convergence.

- support heterogeneity of networks and terminals in a simple way, transparent to the service control
- independency of networks, but possibility to use network specific features as far as needed
- control of services spanning multiple networks
- rapid and easy service development

- easy extensibility for additional networks

3. THE NOVEL APPROACH: NETWORK INDEPENDENT SERVER ARCHITECTURE

To enable network infrastructure independent service definition and control we propose a new service architecture. The architecture defines the components of a control server that resides within several communication networks or that has got access to those. Fig. 1 shows the general concept.

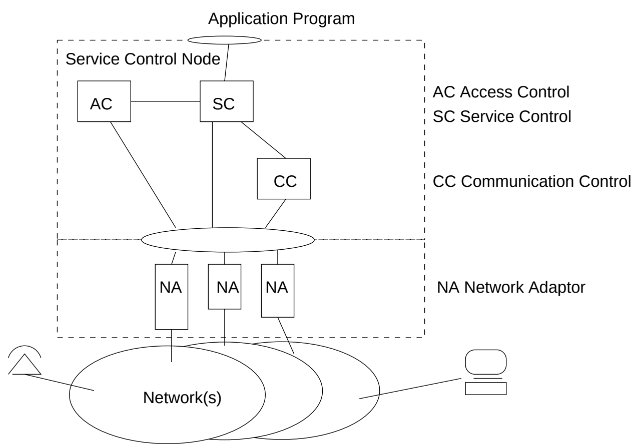

Figure 1 Server architecture general concept

Within the server architecture we distinguish two areas. One part (the lower part in Figure 1) is responsible for the strict separation of the network control and the service control. This is achieved by Network Adaptors (NA). The NAs convert the service control signaling into network specific signaling.

The part that performs the control of the services uses a well-defined interface for the communication with the network adaptors. In particular the service control consists of three main functional blocks. The Access Control components (AC) perform the user access to the server and manage all user specific data. The execution of a specific service is controlled by the service specific Service Control components (SC). These make use of the Communication Control components (CC) to establish communication relationships in the different networks.

The server architecture is open to application service providers, which could provide application specific data for the service execution or which could start a service from within an application specific context.

This architectural concept provides an overall use of heterogeneous networks for the provisioning and control of services. Thereby the operators of such a server need not to be the owners of the networks, but have a point of access to them. Open APIs like an advanced PARLAY API will support the realization of such an architecture. The server could be located at any place within the networks, e.g. close to an IN SCP or at a CATV head end. The more networks to be reachable the better it is. The components of the architecture could also be distributed over multiple servers using a middleware platform.

Services. The server provides its services to all customers in the connected networks. However it is not aimed to offer services that are already present in the different networks (e.g. simple telephony, conference, data transfer), but high-level services that go beyond the functionality of a specific network. These are

- services spanning several networks
- combination of basic services (e.g. conference plus video retrieval)
- separation of access network and execution network (e.g. for VoD)
- supplementary services
- personalized services (e.g. UPT-like one number service, unified messaging)

Section 6 presents a detailed example for typical services.

Networks. The users have access to the server via heterogeneous communication networks. The server accepts requests via signaling channels as well as usage channels; the latter means that the user explicitly addresses the server, connected to the network adaptors. From a network's perspective the server looks like a well-known element. For example in the Intelligent Network we use an SCP based network adaptor. In IP based networks the network adaptor acts as a server (web server, mail server) and could be addressed in this way. Therefore access networks may be e.g. (interfaces in parentheses) POTS (UNI), ISDN / B-ISDN (D-channel or B-channel), IN (SCP), IN-SSP (INAP), Internet (via web server: HTML, WAP, email, etc.), GSM, UMTS, GPRS.

For the establishment of connections or connection-less communication relationships, the server makes use of communication networks. When we separate access networks from communication networks, we have to point out that of course one network may serve for both. In the same way the service control information, which flows between a user and the particular Service Control (SC) during an active service, could be carried on the access network or on other connections that have been set up by the Service Control. In addition to the networks mentioned for access networks further examples for communication networks are DVB, DAB, satellite networks, and CATV. The networks are controlled via standardized interfaces like PINT, PARLAY, JTAPI, JAIN, CGI-scripts, or Gatekeeper, if available or via the user network interfaces.

4. SERVICE CONTROL ARCHITECTURE

So far we have presented the overall concepts of our server architecture. In the following we will describe some more details about the main mechanisms providing the characteristics of the architecture.

4.1. PLATFORM AND INTERFACES

The interconnection of the architecture's components could be set up via any common high-performance signaling network. In general, any distribution of the components is possible then depending on the signaling performance. Since we do not proclaim a new internal signaling network, a subnetwork of existing networks could be used ("virtual signaling network", e.g. high-speed LAN).

The separation of the service control components into the three blocks is based on the session concept of TINA [7] and other similar approaches. Like there we distinguish access session, service session, and communication session. However, the further structuring and functionality differ from the TINA recommendations as can be seen in the following. The main defined interfaces of our architecture result from this separation by sessions. Additional interfaces have been set up for adaptation and supporting components.

The interfaces (see Fig. 2) are defined by the set of messages that can be exchanged with them. All in all we have defined 12 messages to keep the complexity to a minimum. The messages are text based following the SIP concept [6], to support easy message processing.

In particular the messages are:

`ACCESS` starts an access session and delivers information about the current configuration of the user access (terminal, network, NA).

`START` transmits the user's selection to start a service.

`CREATE` The creation of a service session is requested by `CREATE`.

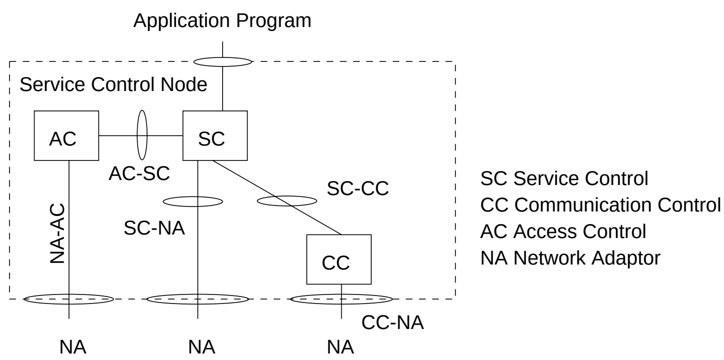

Figure 2 Main interfaces of the service architecture

`INFO` is used for requesting further information exchange between the components e.g. address of invited user. `INFO` requires an answer that is also delivered in an `INFO` message.

`STATUS` In contrast to `INFO`, `STATUS` delivers information without preceeding request, e.g. to store the session status in an user agent.

`END/BYE` Whereas `END` terminates a whole session, `BYE` is used to end the participation of one single user in a session.

`INVITE/SETUP` Both are used for the addition of a new user, a resource, or a connection to a session. `INVITE` is used by the Service Control whereas `SETUP` is a Communication Control message. Both messages are the first of a three way handshake protocol.

`OK` acknowledges a requested communication relationship (`INVITE/SETUP`) or if the requested configuration is not possible it suggests an alternative.

`ACK` is the answer on `OK` and confirms the establishment of the previously reserved configurations.

`REGISTER` is used for the registration of resources and network adaptors.

Infrastructure Independent Addressing. User addressing is performed in two steps. On the one hand each user has got one or more addresses that identify him in each of the different networks connected to the server. At least one address is known to the server from the access network. Further addresses (in different networks) could be obtained when the user connects from different network adaptors or registers addresses in an access session. All addresses are stored with the user profile to be able to contact the user over different networks. The user could configure one main contact address.

For the assignment of the addresses to the corresponding network, we use the addressing scheme: `user@network`. Whereas for `user` the address of the customer in a specific network is used, e.g. a telephone number or an email alias. `network` labels the network or network domain in that `user` is a clear identifier. Valid addresses may be `08928923505@dtag.tel` or `01721234567@D2.gsm` or `kellerer@ei.tum.de`. We have defined this addressing scheme in respect with the addressing of the IETF IP telephony in the SIP protocol [6]. So we achieve compatibility to existing addresing schemes in addition.

For a clear, network independent identification of a user within the server architecture, i.e. among the components, for the identification in different sessions, we use another well-known user indication scheme. Each user is assigned with a `login` and a `password`. In particular both are necessary when a registered user logs on from a previously not used network. The user profile is identified by the login and all user-network addresses are stored there for retrieval within the communication session.

4.2. COMPONENTS

The architecture consists of the area for network adaptation and the area for network independent service control. Fig. 3 shows the decomposition of these two areas in the most important components. Beneath the Network Adaptors (NA) and the components of the three main blocks that represent the session concept, further components are presented that support the latter in fulfilling their functions. These Service Support components (SS) are necessary for adaptation and registration of service and network resources. They will as well be described in the context of the three sessions explained below.

Service Access. The components of the block Access Control (AC) realize the access session and support the user's access to the server (`ACCESS`). The service access fulfils two functions. First to support the user in selecting and starting an information and communication service (`START`). This corresponds to the TINA definition. Second to support user registration and user profile customization (`ACCESS` and `INFO`). In contrast to the TINA definition an access session only lasts until the access connection of a user exists independent from service execution. This means an access session could also be the receipt of an email by the server containing user information (including encrypted login and password).

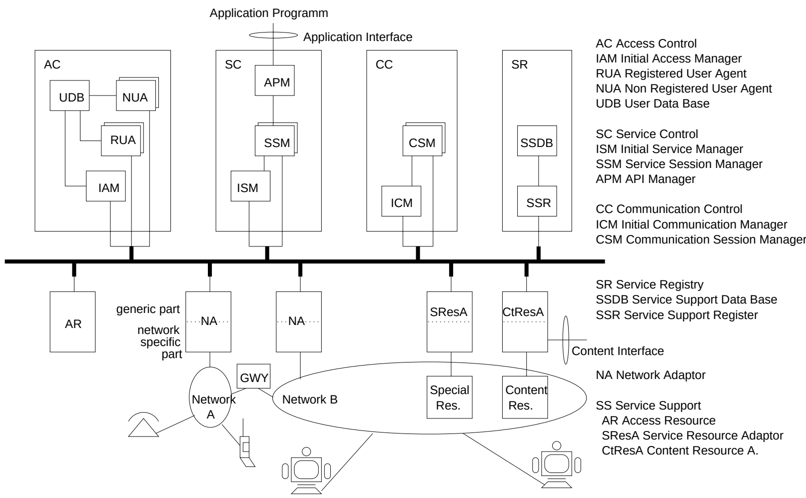

Figure 3 Component decomposition of the server architecture

For the management of user specific data the AC block contains a data base (UDB), which stores a user profile for every registered user with his login. The UDB contains all user data that is relevant for service control, in particular these are

- networks and corresponding addresses, by which the user could be contacted
- terminals the customer uses in combination with the addresses
- the terminal configurations
- the current or, if the user does not permanently update, the last user configuration (network, terminal)
- services for which the user has registered
- parameters for personalized services, e.g. call forwarding configuration
- service independent preferences and configurations of the user regarding QoS, billing

In this way the access session supports *user* as well as *service mobility*, since it supports the user to access services from any network or any terminal.

The User Profile Data Base (UDB) is realized as a directory server using LDAP. Thus a customer could also use his profile for other applications outside the server, e.g. for a browser independent storage of personal bookmarks or for terminal configurations (user specific look and feel).

A so called User Agent (UA) controls the access session for one user on the server and represents the user in all phases of service initiation and execution even beyond the access session (e.g. billing). The UA is selected and instantiated by the Initial Access Manager (IAM) for registered users (RUA). For non-registered users (NUA) a user agent is created for temporary storage and management of the user data.

Access Resource. The Network Adaptors enable the user connection from any network by the transformation of the network specific signaling into the generic messages of the server architecture. For the realization of access functions like authorization and authentication hypertext servers seem to be very useful to set up a convenient front end to the customer. Most terminals are already equipped with a browser. Scripts (e.g. CGI) or applets could control the user interaction with the server components. However there are quite a number of different hypertext servers dedicated to specific situations, e.g.

- HTML server, the most common information server in the internet,
- WML server for WAP clients in wireless communications, and
- VoxML or VXML server for speech based browsing [11].

In addition, many other XML based servers are imaginable in future. Each hypertext language could be transmitted over any network using e.g. HTTP. So it would be a great effort to integrate all servers in each of the network adaptors.

Instead we have separated the hypertext servers in our architecture from the NAs and concentrated them in an Access Resource (AR). The AR provides a centralized hypertext server functionality to all forms of access. This means, the same HTML server, equal if the user logs on via ISDN dial-up (NA-ISDN/PPP) or LAN (NA-LAN) will support every HTML/HTTP based request to the server architecture.

Service Control. When the user has started a service via the access session, a service session is activated (`CREATE`). The central Element of each service session is in analogy to the TINA concept a Service Session Manager (SSM). The SSM controls one instance of a selected service, which has been instantiated with user specific parameters. Created by a central Initial Service Manager (ISM) the SSM is able to communicate directly with the user via a NA to request further information (`INFO`) or exchange control information (`INVITE, BYE, END`). Therefore the SSM uses the NA of the access session or establishes a new connection via the Communication Control (CC).

Service Context. The service session manages a central view of the status of a service, which is represented by the service context. The service context contains user specific information of the access session as well as service specific parts from the service definition. For description of communication relationships defined for a service only qualitative descriptors are used. We have defined for example three QoS classes: asynchronous, relative QoS, and guaranteed QoS. These descriptors are necessary for the Communication Control to select a suitable infrastructure for the execution of the service.

Service Support Components. There are many services who's content is not created by the customers themselves (e.g. telephone dialog) but is delivered by media servers (e.g. video server). These resources are represented in our architecture by the class of Content Resources (CtRes) as a subclass of the Service Support Components (SS). Content Resources could be integrated directly by the SSM into a service (`INVITE`). Beneath that, media server could be treated as normal users if the appropriate control interfaces are not present.

Communication Control. The setup and control of all communication relationships for one service are performed by one Communication Session Manager (CSM) for each service. The CSM maps the qualitative description onto concrete connections and selects the appropriate networks according to the communication part of the session context, which is received from the SSM in the `SETUP` message. The mapping also depends on the user configuration data, which could be obtained from the UDB or the UA (`INFO`). All this information is used by the CSM to decide, which NA to choose to make the best selection in respect with the given situation.

The connection setup is done according to a three way handshake protocol that is repeated in several steps (cf. atomic action protocol of the ITU-T [12, 13]). First the required configuration is requested from the SSM to the CSM (`SETUP`). The CSM selects the NAs and additional service support components and forwards a `SETUP` with concrete data. Each component has to answer the request with an `OK` message either acknowledging the configuration or proposing an alternative. When the CSM has received all `OK`s the configurations are confirmed and activated with `ACK`. If alternative suggestions from the components do not fit the

CSM it starts the procedure again with `SETUP` or has to report to the SSM with an `OK` including alternative solutions.

Special Resources (SRes) e.g. converters or gateways support the Communication Control to provide network independent service execution. The SRes residing in the networks could be controlled similar to the networks by SRes adaptors.

5. PERSONALIZATION AND MOBILITY

Beneath the basic features, which we have described above, the server architecture includes some additional features to support a flexible provisioning of high-level information and communication services.

Mobility: User Agent and Directory Server. Mobility is supported by the centralized management of the user profile including her or his current "location" in a data base (user mobility). Moreover the server architecture allows the access of services from different terminals over heterogeneous networks providing the requirements (QoS, UI) could be met (service mobility).

XML based Service Definition. Each service definition in the server architecture consists of a static and a dynamic part. The dynamic part determines the sequential execution of a session (service logic) in respect with events and the service context. The service context is the service specific part of a session context (see previous section), and contains the static part of a service definition.

Whereas the static part of the service definition, the service parameters, is stored conventionally in data structures, we have chosen a new approach for the service logic description. In contrast to techniques like in the IN, where services only could be defined according to a limited set of building blocks (SIBs), or ISDN/H.323, which have a strict protocol based definition, we propose an approach using a scripting language based on the eXtensible Markup Language (XML). In particular we refer to the Call Processing Language CPL of the IETF IP Telephony Working Group [10] for the definition of our tags. XML based scripting languages provide an easily readable and verifiable way of service definition to support a flexible service development [11]. A simple service definition allows users as well as external providers to use the service architecture for customized services.

Service Discovery: Network Adaptors Registry. An advanced service discovery mechanism supports an online reaction of the server to changes in its environment. For this purpose the architecture includes a Service Registry (SR) in that analogous to the User Profile Data Base available characteristics of network adaptors and adaptors of service support components could be stored.

Along with the approaches of service discovery (e.g. JINI, SLP) components could register with the Service Registry (`REGISTER`) and offer their services. The CSM, when mapping the qualitative description, could retrieve the data from the SR and select appropriate components. Hereby we could also think of selections made upon pricing information and even upon the current traffic situation, which could be registered, too.

6. A MULTIMEDIA SERVICE EXAMPLE

To convey further understanding of the control flow and procedures within the proposed service architecture, a detailed example will be given. We illustrate the universal application of the server architecture with an example based on an advanced voice telephony service that is extended with an information service. The customer Mr. Joe Black is currently on a trip in a car and is able to communicate only via GSM. He wants to contact Mrs. Jane Green, which is working in her office and connected to ISDN and to the company's LAN. We assume both users being registered with the server. The requested services are executed in the following way. Fig. 4 illustrates the steps.

1 Customer Black dials the number of the server and the GSM network adaptor (NA-GSM) accepts the incoming voice call and forwards it to the voice browser Access Resource. The AR begins an access session (`ACCESS`).

2 The user is logging in via DTMF and his RUA (RUAb) is started. Automatically his current configuration is stored.

3 The RUAb offers various services, which he could use with the current connection. The possible selections are presented via speech synthesis by the voice browser.

4 Black chooses *Call "Jane"* and starts his service (`START`).

5 The access session creates the service instance *Call "Jane"* (`CREATE`), which is automatically enhanced with the internal logins of Mr. Black und Mrs. Green. The personal alias "Jane" is transferred for Black into the login of Jane Green. The access session ends.

6 Since the SSM has got all data necessary for the service execution, it requests the establishment of a connection between Black and Green with `SETUP`.

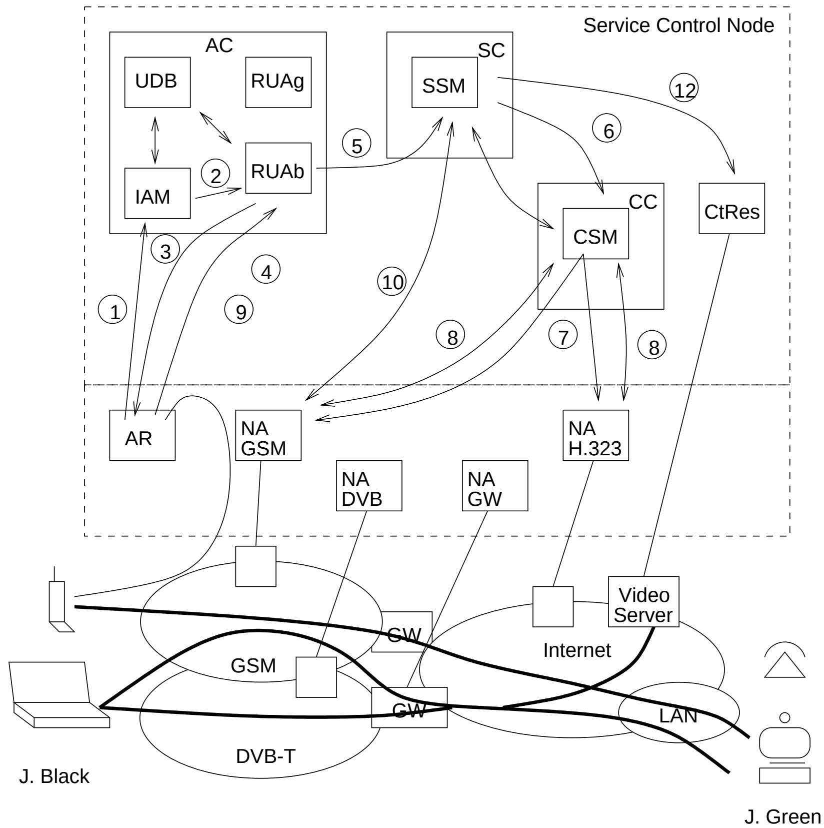

Figure 4 Example

7 The CSM retrieves information about the location of Black and Green and about available networks. The logins are mapped by the UDB to the appropriate network addresses then. At the same time a UA for Green is created to store the current configuration (RUAg). The CSM selects GSM for the real-time connection (guaranteed QoS) for Black and H.323 for Green, since Black prefers an IP based call for lower costs. A H.323/GSM Gateway is available. The control for GSM is done via an advanced PARLAY interface in GSM and for H.323 via a Gatekeeper interface.

8 After all NAs have acknowledged a successful reservation of the connections with `OK`, the connections are finally set up after the `ACK` from the CSM.

After a short voice discussion the users want to consult a document, a video documentation about a product, to get more information about a business decision. Since the GSM connection is not sufficient in bandwidth for Black, he activates his portable PC, which has in addition to another GSM module a DVB-T (Digital Video Broadcast) receiver.

9 User Black starts a new access session; this time originating from his PC via GSM. The access session is now processed via a HTML server (AR). It is identical to the one before (same RUAb) except for the presentation (screen instead of voice). After the login procedures the RUA automatically stores the DVB-T receiver.

10 After Black has activated the service *"Video Retrieval Within Call"* (`START`), the UA checks if this service could be combined with the existing voice dialog. After that the service is integrated within the active service session. The service session takes over the HTML server of the access session and the GSM NA/AR for further message display and the presentation of the video. The SSM requests the input of the video server's address and the chosen video (`INFO`).

11 The further proceeding is similar to the steps described before. But the connections are set up partly in different networks (DVB-T, GSM and Internet, in which the video server is located). For user Green the internet connection remains in use, since the video server itself is connected to the internet.
For both users it is transparent, that the exchanged data is transmitted over different networks and even the interaction channel and the downstream channel are separated. This illustrates the independence of the service architecture from the control of communication relationships within the networks.

12 The video server is known to the server as a content resource (CtRes). Thus the SSM could activate the video server access for Black and Green. The control of the video source is performed by a specific protocol of the video server via the channels that are established by the CSM.

7. CONCLUSION

It has been shown that service provisioning has to consider the increasing heterogeneity of network technologies. Current service architectures are not enough independent from network infrastructure to provide the demanded flexibility for multimedia information and communication services. A feasible solution is the convergence on the service level.

A network independent server architecture has been proposed. Network adaptors in combination with an adaptive selection mechanism in the communication control allow the complete network independent control of services over heterogeneous networks. Networks include telecommunication networks and information networks as well as broadcast networks. The most important advantages of the server architecture are:

- flexible application of the service control and of deployed services in heterogeneous networks with heterogeneous terminals without further changes
- independency from network providers
- adaptive selection of communication networks for service execution
- services can be deployed in networks they were originally not designed for

To complete the architecture some points are still open for further research. For example the integration of pricing within the access session is under investigation. In addition the interaction with some kind of Network adaptors has to be detailed. We have already realized components for an integration of broadcast networks, especially DVB-T [14].

A very critical subject is of course service interactions. We have studied interaction issues for multimedia services and found many new causes for interactions not mentioned in literature yet that will greatly influence the correct behavior of services [15]. Some more strategies for an early avoidance of service interactions will have to be integrated into the server architecture. For the validation a prototype realization of the server architecture is currently under development using UML and SDL in combination with tools.

References

[1] C. Gbgaguidi, J.-P. Hubaux, M. Hamdi, and A. Tantawi. A Programmable Architecture for the Provision of Hybrid Services. IEEE Communications Magazine, Vol. 37, No.7, July 1999, pp. 110-116.

[2] E. Holz, O. Kath, M. Geipl, G. Lin, and V. Vogel. The CAMOUFLAGE Project – Introduction of TINA into Telecommunication Legacy Systems. In Proc. of TINA'97, Novemver 1997, pp. 206-215.

[3] D. Guy. TINA-IN-Workgroup: Intelligent Networks Migration Towards TINA. 1999. Http://www.tinac.com/wg_sig/in/Main.htm

[4] W. Kellerer and B. Quendt. Multimedia Service Architectures. In Proc. of EUNICE'98, Munich, Sept. 1998, pp. 67-76.

[5] M. Korpi and V. Kumar. Supplementary Services in the H.323 IP Telephony Network. IEEE Communications Magazine, Vol. 37, No. 7, July 1999, pp. 118-125.

[6] H. Schulzrinne and J. Rosenberg. Internet Telephony: architecture and protocols – an IETF perspective. Computer Networks, Vol.31, 1999, pp. 237-256.

[7] TINA Service Architecture, Telecommunications Information Networking Architecture Consortium, Http://www.tinac.com, 1997.

[8] PINT – PSTN To Internet Integration. IETF. Http://www.ietf.org /html.charters/pint-charter.html, 1999.

[9] Parlay - the API for Secure and Open Access to Networking Functionality for Third Party Applications outside the Network. In Proc. of: ICIN'2000, 6th International Conference on Intelligence in Networks, Bordeaux, January, 2000, pp. 17 - 21.

[10] J. Lennox, H. Schulzrinne. CPL: A Language for User Control of Internet Telephony Services. Internet Draft, Internet Engineering Task Force, IPTEL WG, February 1999. Work in progress.

[11] B. Guedhami, W. Kellerer and C. Klein. Web enabled Telecommunication Service Control using VoxML. In Proc. of SmartNet 2000, Vienna, Sept. 2000.

[12] H. Müller. Flexible Signalisierungsarchitektur für Multimediadienste mit heterogenen Endgeräten. Dissertation, Technische Universität München, 1996.

[13] ITU-T Recommendations X.851/2. Information Technology – Open Systems Interconnection - Protocol For The Commitment, Concurrency And Recovery Service Element. ITU-T, 1993.

[14] P. Sties and W. Kellerer. Radio Broadcast Networks Enable Broadband Internet Access For Mobile Users. In Proc. of EUNICE'99, ISBN 84-7653-717-4, Barcelona, September 1999.

[15] W. Kellerer, P. Sties, and P. Moritz. Service Interactions beyond IN: The new Challenge for Multimedia and Convergence. In Proc. of ICIN'2000, 6th International Conference on Intelligence in Networks, Bordeaux, January 2000, pp. 277-282.