PROTOTYPING A PDA BASED COMMUNICATION APPLIANCE

R. Ackermann[1], M. Görtz[1], M. Karsten[1], R. Steinmetz[1,2]
[1] Darmstadt University of Technology
Department of Electrical Engineering and Information Technology
Industrial Process and System Communications - KOM
[2] Fraunhofer Institute for Integrated Publication and Information Systems - FhG IPSI
Darmstadt - Germany
{Ralf.Ackermann, Manuel.Goertz, Martin.Karsten, Ralf.Steinmetz}@KOM.tu-darmstadt.de

Abstract: *PDAs have become an accessory that a large community uses for managing their personal data. With the development of new devices with additional network and audio capabilities they gain the potential to be used in more comprehensive communication scenarios too. On the example of usage for IP telephony we show, how new applications must not be viewed in an isolated way and need innovative approaches for both system hard- and software enhancement as well as for the integration with heterogeneous infrastructure components. The challenging task for creating so called communication appliances is not just to transfer existing desktop mechanisms to smaller and mobile computers but to consider their specifics and establish additional architectures and mechanisms that meet those best.*
Our paper describes the prototyping of a (wearable, voice-) communication device based on an off-the-shelf iPAQ PDA, design and usage experiences and potential future enhancements.

KEYWORDS: Ubiquitous Communication Services, IP-Telephony, PDAs, Wearable Computing, Component based System Design

1 Introduction

1.1 Starting Situation and Motivation

In recent years communication paradigms evolve very fast and we experience a major shift from fixed and stationary systems towards nomadic and mobile communication devices. A realistic future scenario comprises nearly ubiquitous network access in large areas of the world. Moreover the rapid transition towards an open and flexible IP-based heterogeneous network infrastructure will accelerate the definition and deployment of various services spanning the range from speech over "conventional" to multimedia applications.
Additionally we also face a change in the type of devices that are used for managing data and accessing services. Whereas computational intensive applications were traditionally bound to host systems or desktop computers, the recent generation of PDAs introduces a shift in this assignment of functions. In addition to having easy means of accessing and storing data, one of the basic motivations for using those portable devices is "(always) being able to communicate" (as shown with the tremendous success of portable phones). With the recent state of technology it gets possible and desirable to use PDAs (and their future successors) for not just managing notes, contacts and calendars but for also accessing a huge number of infrastructure based services and for even telephony applications.

1.2 Organization of the paper

The paper is organized in the following main parts. After a introduction of the concept of communication appliances and basic IP telephony mechanisms we list our requirements for establishing a heterogeneous scenario that especially considers the use of PDAs as end systems. The discussion of alternative designs, that differ in the way they use specific protocols and split functionality between infrastructure and end system components, is followed by a description of the implementation that we were working on. We show how our initial goals were met and list experiences gathered during development and test. Finally we conclude our paper and give an outlook on the future potential of our approach and prototype.

2 Basics and Related Work

2.1 Communication Appliances

In the past we mainly find a number of applications – each packaged in a kind of a ”black box” that are specialized for distinct purposes. You are e.g. using a cellular phone for audio communication whereas the addresses and phone numbers of your communication partners are stored on another device.
With the emerging use of these PDAs and other innovative computer end systems the new term ”communication appliance” has been introduced. It (though used with several and not fully consistent meanings) describes the combination of both hard- and software components as tool for personal use (its helpful to think of a ”Swiss Army knife” for an analogy). To fulfill the demand for multimedia communication, everywhere and at anytime, communication appliances have to overcome the apparently contradictory and incompatible requirements of being small and multi-purpose, power-efficient and powerful, autonomous and interacting with heterogeneous infrastructures. We will show how many of these specifics apply to and have to be solved in the area of using PDAs for IP telephony.

2.2 IP Telephony as challenging Example Scenario

IP telephony forms a promising technology to provide enhanced and new communication and application scenarios to a number of users. Basically every networked computer system with appropriate audio capabilities can potentially be used to communicate with other IP-based but via gateways also conventional telephones. In combination with Presence and Instant Messaging applications this holds an enormous potential for the support of seamless ubiquitous communication.
In addition to the support of a sufficient audio quality by means of an appropriate treatment of the RTP streams carrying the media data, the major challenge in this area is, how to design and deploy a powerful, robust and though efficient signaling for setting up connections, providing the expected feedback (e.g. ringing, busy, called party hung up) and negotiating and (ex)changing the media transport characteristics (ports, codecs). For that purpose we currently see two main signaling protocols H.323 [16] and SIP [14] evolving in parallel. Since the coexistence of both protocol families is expected for a longer future period, communication partners using both of them should be supported in a realistic scenario. This can basically be done using appropriate gateways.

2.3 Related Work

Current innovations in the area of ubiquitous and mobile computing [10][15] are pushed from research both on the (hardware) system as well as the protocol and application side. The progress in technology and the availability of powerful hard- and software platforms recently enabled the development and practical deployment of a number of applications that try to use small devices [23] in a collaborative manner [20][19]. To fasten system development and to allow for the decomposition of services there are a number of activities for developing powerful middleware [11]. For our prototype we benefit from the work that is currently done for providing Linux and other Open Source solutions for off-the-shelf PDAs as the iPAQ [4] (and have actively contributed to these efforts). Finally the work in the IP telephony area relates to current initiatives that are coordinated by the IETF iptel Working Group and a number of very active university [9] and industry parties [12].

3 System Architecture and Design Considerations

3.1 Targeted Scenario

Our intended scenario compromises a set of mobile users with PDAs that are connected to an infrastructure via Wireless LAN and are able to both originate and receive voice calls and other messages. It intentionally tries to support heterogeneous scenarios (with a mix of possible IP telephony signaling protocols to be used by the communication partners) and the demand of flexibility for future enhancements regarding both the system hardware and IO-interfaces as well as additional applications.

3.2 Requirements and bounding conditions

Out initial system requirements are listed in Table 1.

Requirement	Implication(s)
Voice Connectivity	integration of mechanisms to ensure interoperability with existing IP telephony applications and devices
Consideration of device specifics	small memory footprint (both for storage and execution), technical feasibility considering processor power and specifics like e.g. missing processor support for floating point arithmetics
Scalability	support for the potential parallel usage of infra-stucture based mechanisms by a multitude of end systems
Code re-use and easy development and deployment	usage of a uniform code base, possibility for cross-developing and -testing components and their interaction

Table 1: Prototype Requirements

In order to meet those requirements we did an evaluation and pre-selection of base components which has been based on the criteria: development and runtime platform (Operating System, implementation language and portability), availability of program source code for basic functionality (IP telephony signaling stacks, User Agents), potential inter-operabilty, estimated complexity of integration with other components, usage conditions (Open Source vs. evaluation vs. commercial only versions). This lead to the selection of Linux as preferable development and runtime platform.

3.3 Service Signaling, Provisioning and Location Alternatives

Whereas there is a common agreement and well-established practice that audio media streams are transmitted using RTP and preferably directly between the communicating systems, there are a number of choices concerning the signaling. As a very straight-forward approach it would be thinkable to just try to re-use existing desktop applications and compile them for the target devices. When doing initial experiments on porting as well H.323 as SIP software we have found though, that (though compilation

succeeded) this (in the H.323 case) results in software that does not meet the device characteristics because of the resulting unacceptable memory footprint. The H.323 terminal program OpenH323 ohphone and the necessary shared libraries (cross-)compiled for a Compaq iPAQ PDA (StrongARM processor, 32MB RAM, 16MB Flash) running Linux have a size of 9236292 bytes. A basic SIP user agent for signaling plus the RTP component for media streaming can be implemented considerably smaller due to its lower complexity [21].
We show alternatives to just re-using existing software in an unmodified manner now. Communication relationships within a voice call or conference scenario are typically end-to-end. This does not necessarily map to a direct mapping of the used signaling and media streams though. There is a number of alternatives, especially

- end-to-end signaling and media streaming using one or a number of supported protocols (this corresponds to the shown inappropriate ”placing all the functionality at the client” approach)
- direct or indirect ”routing” of media streams, whereas the signaling streams are generally routed and processed via additional specialized instances (e.g. gateways / translators or proxies).

The later case which has the potential to reduce the complexity and the number of necessary supported mechanisms in the end systems has a number of subtypes. We can either

- fully or partly translate the signaling protocol syntax and semantics by means of signaling *gateways*, thus doing the signaling to the end-system (on the so called ”last link’) using one (preferably the less complex one) of the protocols that we generally use in the system

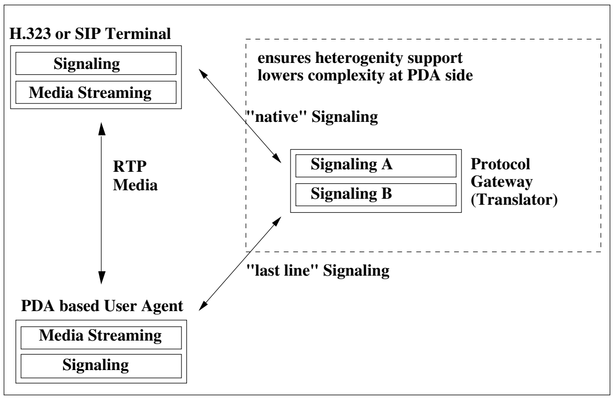

Figure 1: Gatewaying approach with direct RTP-traffic

- terminate the signaling in a (single- or multi-protocol) *proxy* that becomes part of the infrastructure and just uses a very limited signaling towards the end-system itself

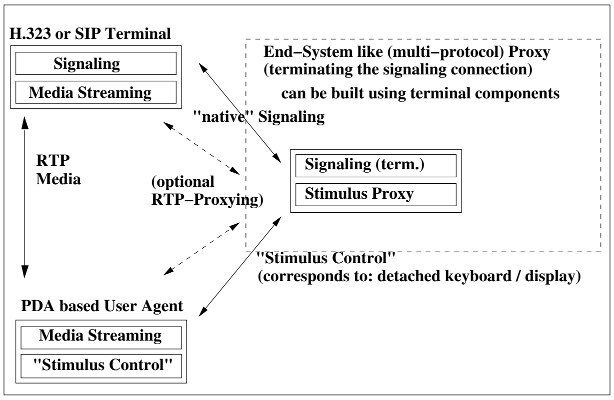

Figure 2: Proxying approach

The first approach has already been considered and implemented [22][8] in a prototype targeting at the general use within a multi-protocol signaling (backbone) architecture. It is helpful in our PDA based scenario as well because it allows to reduce clients complexity at the tolerable cost of handling protocol translation in the gateway.
The proxy approach is especially well-suited for our purpose of integrating small end systems. Those become decomposed in a way that their main part gets located in the common infrastructure and just a very small part (we are just transmitting the keypad press and on-hook / off-hooks events as well as the resulting operation like ringing or text to be displayed) towards the phone. This can be done by means of a standardized stimulus protocol as currently under consideration by the ITU standardization bodies [17] for usage with small, feature-limited terminals. We are using a proprietary ASCII-based protocol though at the moment. The RTP media streams have to be transmitted to the end system in any case and will only be proxied if necessary (as a straight-forward though not ideal solution) for passing firewalls that have problems to cope with dynamically negotiated RTP ports for the media streams.
There are a number of alternative mechanisms for associating an end-system and its identity (e.g. by means of an E.164 number or a symbolic name like a SIP-URI) with a gateway or proxy that is responsible for it. At the moment we use a static preconfigured association where the care-off systems address is stored in the clients runtime configuration files. The care-off system itself is responsible for (proxy-)registering the client with the underlying infrastructure (by means of a SIP REGISTER message or H.323 RAS Registration Request RRQ message). In combination with dynamic IP address configuration using DHCP at the client side, this allows for a certain flexibility and terminal mobility already. It holds potential for future enhancements though. At the moment we are investigating the use of Service Locating Mechanisms (preferably using the Service Location Protocol SLP [13][18]) that also allow the clients to identify and autoconfigure themselves in an unknown environment that they move into.

4 The Prototype System

4.1 Computing and Communication Plattform

The Compaq iPAQ is an off-the-shelf PDA that uses a StrongARM processor running at 206 MHz. It follows a modular design with a core unit having a color-graphic 320x240 touch-screen interface, an integrated microphone and speaker, an IrDA port for SIR or FIR serial communication and an external connector that provides both the signals of the iPAQs serial port as well as an USB interface (the device is an USB slave not a master though). Its main interfaces for external devices is a PCMCIA ”jacket” that enables the use of storage cards (e.g. with an IBM microdrive you reach 1GB storage capacity) as well as ethernet and wireless network cards. The device is initially sold with Microsoft PocketPC software but due to the Operating System being stored in the systems Flash Memory, it can be updated to be used with other Operating systems. These specifics and the complete packaging fasten the time for enhancements because we do not have to develop a whole new system from scratch.

4.2 Hard- and Software Enhancements

Compaq made the specifications of the interfaces available to the public and actively supports the further development of Linux on the device. [4] has detailed instructions on how to flash the original OS with Linux and develop for that system. Programming can either be done on an ARM system (such as the iPAQ itself when enhanced with a microdrive providing enough storage capacity) or using a cross-compile toolchains.
Figure 3 shows the schematics of the core system with the enhancements that we have developed for usage in our scenario. For an in-detail description of the software and hardware components we refer

to the referenced URLs where all the necessary schematics and code can also be downloaded.

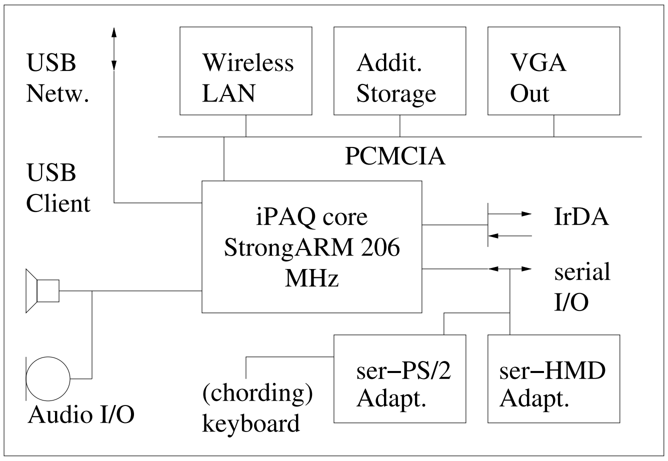

Figure 3: Components of the Enhanced Prototype System

Our enhancements include

- a micro-controller-based serial to PS/2 input adapter for external (chording) keyboards [6]
- a PCMCIA-based standard VGA interface with X-support for attaching Head Mounted or standard Displays [3]
- an external display interface that is based on a video (PAL/NTSC) character generator that is controlled via the serial port [1]

The systems USB client port can be used to establish IP networking with another USB host (such as the majority of x86 based systems), thus combining the device with other components in a ”body-area-network”. This approach has successfully been tested in combination with a AMD ELAN 520 based Wearable Computer [5]. The several parts of the hybrid system provide the functionality that they are specialized on or can provide most energy- or cost-efficient.

5 Implementation Results and Experiences

5.1 Established System and Features

The prototype system allows to place or receive calls while moving inside our WaveLAN based wireless infrastructure. Interoperability has been tested with a number of heterogeneous IP telephony clients and infrastructure components using both H.323 (Microsoft NetMeeting, Linux ohphone, Siemens LP 5100 ”hard-phone”, OpenH323 opengate gatekeeper) and SIP (VOCAL ua User Agent, Pingtel xPressa ”hard-phone”, Columbia University SIP server sipd). The iPAQ audio hardware allows for full-duplex audio communication (due the using Linux, whereas full-duplex support when using PocketPC is still missing) and an acceptable end-to-end delay. In individual tests call setup times could not be distinguished from reference scenarios with stationary desktop based ”soft-phones”.

5.2 Prototype Evaluation

We initially formulated the design and evaluation criteria for our prototype system. Its implementation and test showed a number of both quantitative as well as qualitative results.
The memory footprint of the resulting binaries (when using the SIP-Gateway but especially the infrastructure-based approach with stimulus signaling) is considerably smaller. Whereas the OpenH323

based H.323 binary is 9236292 bytes large, both basic SIP signaling as well as the stimulus based variant were implemented with less then 1000 lines of Tcl/Tk [7] code for the signaling. The gateway or proxy are just responsible for the signaling and can be used for multiple communication relationships in parallel without forming a performance bottleneck. We used a standard RTP library [2] that can be licensed for free for research purposes for media streaming in these variants.
Due to the usage of a standard Linux environment (both for development and runtime) the approach gains a high amount of code re-use, flexibility, extendability and possible combinations with other standard components. This must especially considered when comparing against single purpose, specialized embedded systems that are still typically used when building (consumer) communication products. This also allows for decreased monetary costs and a reduced time to market.

6 Conclusion and Future Work

The presented prototyping environment is a perfect-sized platform for rapid development of new communication services. Using it we could successfully show the feasibility of our decomposition and infrastructure integration approach, that forms a general paradigm for enhancing the functionality of small devices, in a working environment. The paper does not cover the promising area of integrating Value Added Components such as Presence Services or Instance Messaging yet. We are going to evaluate those, that especially benefit from the availability of our Linux based prototype, on top of the existing prototype at the moment.
All the schematics for the hardware extensions to the iPAQ as well as the software that we develope(d) and used are freely available from the URLs shown in the reference section. We intend to present the systems prototype and future results of our on-going enhancements at the conference and will support other parties using them as research or reference platform.

REFERENCES

[1] *An iPAQ based Wearable Computer*. http://www.iptel-now.de/PROJECTS/WEARABLE/IPAQ_WEARABLE/ipaq_wearable.html.

[2] *Bell Laboratories RTP Library*. http://www.bell-labs.com/topic/swdist/.

[3] *Giving presentations with a Compaq iPAQ running Linux*. http://www.iptel-now.de/HOWTO/IPAQ_PRESENTER/ipaq_presenter.html.

[4] *Linux on the Compaq iPAQ PDA*. http://www.handhelds.org.

[5] *MIP 520 based Wearable Computer*. http://www.iptel-now.de/HOWTO//MIP520/mip520.html.

[6] *PS/2 to serial serial converter with Linux input support*. http://www.iptel-now.de/HOWTO/PS2SER/ps2ser.html.

[7] *Tcl/Tk for the iPAQ running Linux*. ftp://ftp.handhelds.org/pub/apps/tcltk/.

[8] Ralf Ackermann, Vasilios Darlagiannis, Manuel Goertz, Martin Karsten, and Ralf Steinmetz. An Open Source H.323-SIP Gateway as Basis for Supplementary Service Interworking. *In Proceedings of the 2nd IP Telephony Workshop, New York*, pages 169–175, April 2001.

[9] Ashutosh Dutta, Faramak Vakil, J.C Chen, Miriam Tauil, Shinichi Baba, and Henning Schulzrinne. Application layer mobility management scheme for wireless internet. In *3Gwireless 2001*, page 7, San Francisco, May 2001.

[10] A. Helal et al. Any Time, Anywhere Computing; mobile computing concepts and technology. In *Kluwer Academics Publisher*, Boston, 1999.

[11] Stephen Fickas, Gerd Kortuem, and Zary Segall. Software Organization for Dynamic and Adaptable Wearable Systems. *International Symposium on Wearable Computers*, 1997.

[12] Stefan Gessler, Oliver Haase, and Andreas Schrader. A service platform for Internet telephony. In *Proceedings of the 1st IP-Telephony Workshop (IPtel 2000)*, Berlin, Germany, April 2000.

[13] E. Guttman, C. Perkins, J. Veizades, and M. Day. Service location protocol, version 2. *RFC 2608*, June 1999.

[14] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg. SIP: Session Initiation Protocol. *RFC 2543*, March 1999.

[15] R. Hull, P. Neaves, and J. Bedford-Roberts. Towards Situated Computing. *Proc. of IEEE First ISWC*, pages 146–153, October 1997.

[16] International Telecommunication Union. H.323 - Packet based Multimedia Communication Systems. *Series H: Audiovisual and Multimedia Sytems, Telecommunication Standardization Sector of ITU, Geneva, Switzerland*, 1998.

[17] International Telecommunication Union. Annex L to Recommendation H.323 - Stimulus protocol. *Series H: Audiovisual and Multimedia Sytems. Telecommunication Standardization Sector of ITU, Geneva, Switzerland*, 2001.

[18] J. Kempf and J. Rosenberg. Finding a SIP Server With SLP. *Internet Draft draft-kempf-sip-findsrv-00.txt*, February 2000.

[19] B.A. Myers, H. Stiel, and R. Garggiulo. Collaboration Using Multiple PDAs Connected to a PC. *Proc. of the ACM Conference CSCW*, pages 285–294, November 1998.

[20] T-L. Pham, G. Schneider, S. Goose, and A. Pizano. Composite Device Computing Envronment: A Framework for Augmenting the PDA using surrounding Resources. *Proc. of the ACM CHI 2000 Conference Workshop Situated Interaction in Situated Computing*, April 2000.

[21] H. Schulzrinne and J. Rosenberg. A Comparison of SIP and H.323 for Internet Telephony. *The 8th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV 98), Cambridge, England*, July 1998.

[22] K. Singh and H. Schulzrinne. Interworking between SIP/SDP and H.323. *1st IP Telephony Workshop 2000, Berlin*, April 2000.

[23] M. Weiser. Hot topic: Ubiquitous computing. *IEEE Computer*, pages 71–72, October 1993.