Managing Personal Communication Environments in Next Generation Service Platforms

Ralf Kernchen[1], Matthieu Boussard[2], Cristian Hesselman[3], Claudia Villalonga[4], Eric Clavier[5], Anna V. Zhdanova[6], Pablo Cesar[7]

Abstract—**The current access to mobile services a user has, is defined by the user's mobile terminal as the single entry point to an operators network. This comes along with a set of limitations. Although performance and multimedia capabilities of mobile devices are constantly increasing, user-service interaction is still limited due to physical constraints imposed by the form factor. Another drawback is the varying ability of devices to download and execute new services. At the same time it is not possible to synchronise, exchange or share the user's data and media content among different devices.**

In order to overcome these limitations this paper presents the concept of the Distributed Communication Sphere and the according architectural framework that allows its management. This framework defines functional components to enable multi-device delivery of communication and media service sessions, user input interpretation, terminal management and resource discovery. It also provides flexible service delivery through the dynamic desktop component and relies on intelligent service enablers of the underlying service platform architecture, such as context-awareness, service provisioning and personal profile enablers.

The work has been performed as part of the EU IST-SPICE (027617) project targeting intelligent extensions for next generation service platforms.

Index Terms—**next generation service enablers, multimodal interfaces, multi-device environments, context-awareness,**

I. Introduction

With the accelerating proliferation of networked multimedia and communication devices and their increasing functional richness, users can easily feel overwhelmed by the complexity imposed to manage their interoperability for certain user interaction needs. Proposals for the intelligent integration of such devices have been researched in relation to pervasive environments [1, 2] for example. However these efforts do not apply easily to mobile environments and their specific requirements. Typically, mobile users access their services in a constantly changing environment, so that user interfaces for user-service interaction, as can be composed from the available devices, changes frequently. The field of multimodal user interfaces in general provides the concepts for a more natural user interaction by allowing the integrating of modality user input from two or more input modes (e.g. speech, gestures, gaze, etc.) in combination with the coordinated presentation of multimodal output [3]. At the same time people tend to interact in a more multimodal fashion if tasks are getting complex [4]. However, the current state of the art shows some needs for further research, when it comes to the challenge to apply multimodal user interaction to changing mobile multi-device environments, as addressed in this paper. Initial work has been performed on this, as example for user interface adaptation for mobile terminals as described in [5].

Another limitation is imposed by the trend of user created content and media content in general, which usually is stored in a distributed fashion amongst different devices. The sharing, exchanging and synchronisation of the content in the different environments can require huge efforts, especially when users want to do it manually. Hence it should be supported automatically and optimally by the mobile operator platform.

The third drawback comes from the fact that service software updates and new services download and execution in general are only supported in a limited fashion. Proposing new services based on the changing context situation such as the location of the user is not supported by the current functionalities at all.

Therefore to overcome the limitations in the mentioned areas of user-service interaction in multi-device environments, user data synchronisation and dynamic service updates, the paper proposes an architecture approach namely the Distribute Communication Sphere (DCS) management system.

The paper is structured as follows. Section II provides a summary of challenges that arise when addressing the limitations on multi-device user-service interaction, user data management and dynamic service updates. Based on this section III introduces the DCS management architecture addressing the challenges. Characteristic use cases for the

Manuscript received January 31, 2006. This work was supported in part by the European Union Information Society Technology in the Sixth Framework Program under the contract number IST-027617 (SPICE project).
[1]University of Surrey, Guildford, Surrey GU2 7XH, United Kingdom (phone: +44-1483-683-423; fax: +44-1483-686-011; e-mail: r.kernchen@surrey.ac.uk).
[2]Alcatel CIT, France (e-mail: Mathieu.Boussard@alcatel-lucent.fr).
[3]Telematica Instituut, Netherlands (e-mail: cristian.hesselman@telin.nl).
[4]NEC Europe Ltd., Germany (e-mail: claudia.villalonga@netlab.nec.de).
[5]France Telecom R&D, France (e-mail: eric.clavier@orange-ftgroup.com).
[6]University of Surrey, United Kingdom (e-mail: a.zhdanova@surrey.ac.uk).
[7]Centrum voor Wiskunde en Informatica, Netherlands (e-mail: p.s.cesar@cwi.nl).

defined components are given in section IV to highlight the dynamic aspects of the system.

II. Challenges

The introduction section highlighted the limitations and gave some initial thoughts on how to potentially overcome them. Nevertheless a number of challenges arise, which need to be identified before proposing a solution. In the following the challenges are subsequently described categorised by the objective.

A. *User-service interaction in multi-device environments*

Human computer interaction (HCI) probably is the research area with the most commonalities to this high level challenge. Specifically concepts provided in the field of multimodal user interface research should be applied for user-service interaction. However, not all particular problems in this area have been solved yet and concepts need to be extended for multi-device environments. The following list provides the main challenges to be addressed in relation to this:

- Discovery of user interaction resources, such as devices which can be used for multimodal user input or media output representation, with their according provided modality services (e.g. microphones for speech input or displays for html rendering)
- Coordination of service and communication session media output in a structured manner in order to provide users with the best means of media presentation for a given set of available modality devices.
- Interpretation of user input is gathered from different devices and usually referred to as multimodal fusion. Users should be enabled to interact with the service through different modality input channels, which should be dependent on the service specific usage evaluated in input by input fashion or in a coordinated manner.
- User-service sessions should be constantly re-evaluated based on resource and context changes.

B. *User data synchronisation*

The synchronisation of user data in general maintains the coherence of a collection of user data sources (e.g. media content, address book, user terminal configuration parameters). The complexity of the task demands that it should be automated to be fully exploited. The following challenges are intrinsic:

- Propagation of user data modifications among user's devices.
- Intelligent conflict management in case of separate modification of data sources.
- Ensure access to up-to-date data for users even if devices are not connected to the platform.
- Take into account certain constraints imposed by the current user context (e.g. network bandwidth or costs), also based on user defined preferences

C. *Dynamic service updates*

Dynamic service updates address the need for updates on services and service representations on the user's terminal based on his current situation, in accordance with the operator's service platform. New services may be proposed and non relevant services need to be set passive or even deleted. Automatic service updates, especially when including user input dialogues, provide a number of general challenges:

- Decoupling of the service logic for the service platform side and the user interface part of the service.
- Rendering of service user interface on the terminal device
- Integration of the service logic with the user-service interaction paradigm.
- Providing generic dialogue templates for platform and service alerts.

Providing these high level challenges, the next section introduces the Distributed Communication Sphere managements system addressing the challenges.

III. Distributed Communication Sphere (DCS) Management Architecture

The goal of the DCS Management architecture is to overcome limitations on multi-device user-service interaction, user data management and dynamic service updates in mobile environments and to address the given challenges in an integrated and consistent manner. Moreover the link to the operator's service platform has to be taken into account.

The Distributed Communication Sphere (DCS) concept itself defines entities which are an intrinsic part of the user's communication. As such it collects information about and represents resources relevant to the current user-service interaction. These resources are not only devices or sensors, but can be also the available networks, services, buddies or groups the user belongs to. The DCS forms a 'sphere' around the user and is typically subject to frequent changes, for instance as a result of user mobility.

In order to manage this complex setup, the DCS management architecture defines a well selected set of components as depicted in Fig. 1. The illustration shows the high-level organisation of the four main components, which are briefly described in the following list:

- The *Dynamic Desktop*, provides a graphical user interface metaphor for the mobile terminal as being a flexible portal to the user's mobile services, provides information about the user's DCS and allows control of the current multi-device service interaction. Further it organising service items called widgets of the user subscribed services in a flexible and easy way for the user.
- The *Resource Discovery System*, which dynamically discovers the available resources in a user's DCS.
- The *Multimedia Delivery and Control System*, which can dynamically change the mode of interaction of a service (e.g., from speech to gestures) and is also capable of moving parts of a service between different devices.
- The *Terminal Manager*, which enables the devices in a user's DCS to share (synchronise) data.

These components are integrating with the current user context situation by making use of a “Knowledge Layer” as also shown in Fig. 1. This mechanism enables them to retrieve context information about a particular user through the Knowledge Management Framework (KMF) [6], retrieve the user's preferences, and obtain recommendations for best user interaction means (e.g., about which modes of interaction or which devices to use).

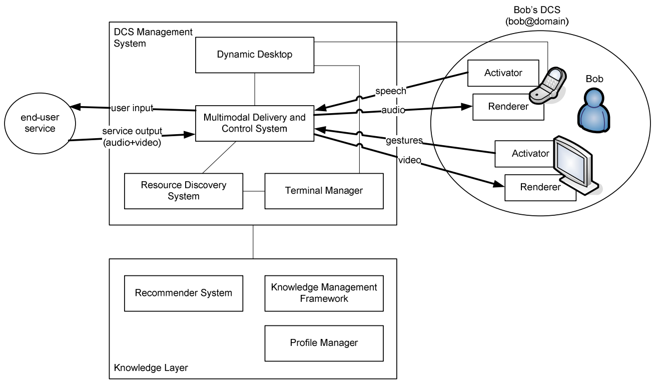

Fig. 1. High-level organisation of the DCS Management System.

The remainder of this section subsequently discusses the Dynamic Desktop, the Resource Discovery System, the Multimodal Delivery and Control System, and the Terminal Manager as part of the DCS managements system in more detail.

A. Dynamic Desktop

The Dynamic Desktop as indicated earlier provides a flexible portal to the user services provided by the operator service platform and user configuration functionalities. For this it displays the list of end-user services (represented by dynamic widgets) on the user's mobile terminal that users have previously subscribed to and dependent on the current context are available. Further it provides a graphical representation of the DCS resources and a service browser and locator in order to search for services the user might want to add to the dynamic desktop.

User can install, update, and remove service widgets from the dynamic desktop. These widgets provide different views on the dynamic desktop in order to show the most relevant information in the right size. The collapsed view shows only a service icon, the focus view shows relevant and dynamically updated service information in a compressed fashion and the full view opens a full screen interaction view of the service widget. The service platform can based on the current situation of the user activate, freeze or hide service widgets in order to always keep the desktop up to date and relevant for the current situation. This mechanism is support by a widget manager subcomponent.

Moreover services and the platform, can notify the user about occurring relevant changes using predefined dialogue templates. Based on the preferences the user has defined the system can send pure information messages or show a dialogue where the user can select certain parameters (e.g. which device to use, when choices are ambiguous or a new service might be interesting for the user). These messages are always carefully selected by the initiator in order to provide as much information as necessary, with the least intrusive means.

B. Resource Discovery System

The Resource Discovery System (RDS) is responsible for probing the user's DCS to provide information about discovered resources in terms of devices, networks and services. It receives information about resources that should be added to the user's DCS, either because they have been directly discovered by the user terminal through discovery mechanisms such as Bluetooth SDP, or because platform components decided to add the resource based for example on user context information, such as location. RDS tries to find a Knowledge Source component associated with this resource; if no such component is available in the platform, it is created. The reference to this Knowledge Source is then added to the user's Knowledge Agent [7], which maintains the list of such references relevant to the DCS of the user at a given time. Other platform components such as other DCS Management System blocks can then subscribe to the Knowledge Agent and the Knowledge Sources it contains in order to get notified of relevant changes in the DCS, such as the availability of a certain network or device. The state of the DCS can be reflected through a dedicated Dynamic Desktop widget.

C. Multimodal Delivery and Control System

The Multimodal Delivery and Control System (MDCS) enables users to interact with a service in various user-service interaction ways (e.g., by means of touch, gestures, or voice) and to seamlessly move from one modality to a different one, while maintaining the current service session. To accomplish this, the MDCS continuously determines the most appropriate modality for each of the services with which a user interacts and then switches to that modality. Similarly, the MDCS also determines which devices in a user's DCS are most appropriate for the interaction with a particular service and then uses these devices and their provided modalities for the user-service interaction. The MDCS may decide to use multiple devices to collect input from the user (e.g., a microphone for voice input and a touch screen for gestures) or use multiple devices to render the output of the service (e.g., a wall-mounted display for video output and a home stereo set for delivery of audio).

The MDCS distinguishes input and output modalities. The *input modality* of a service is the way in which that service accepts inputs from the user. Examples of input modalities include mouse clicks, gestures, voice commands, and keystrokes. The input modality of a particular service may also consist of a combination of these primitive modalities. This process is known as multimodal fusion [3]. Modality input in the DCS management system is represented by so called activators which do the recognition of the user input and translate them in a modality input representation. An example of input fusion is when a user fast forwards in a service's audio-video content by means of a combination of gestures and voice commands (e.g., by saying "fast forward" and at the same time point at the scene to fast forward to).

The *output modality* of a service is the way in which the SPICE platform renders the output of a service. Examples of output modalities are audio, video, and text. As with input modalities, the output modality of a particular service may be a combination of these primitive output modalities (e.g., audio and video). This is referred to as modality renderers.

The MDCS can change the input and output modalities of a service independently for each user that it serves. The most appropriate modality for a particular service (input or output) depends on the preferences of the user, his context (e.g., if he is walking or driving a car), on the resources available in the user's DCS, and on the limitations that a service provider might put on the multimodal delivery of its services (e.g., limit the delivery of a service to certain devices that have large enough displays). A change in one of these parameters might lead to another modality becoming more appropriate.

The MDCS may also stick to the same modality for a particular service, but move to a different set of devices to interact with the user. For example, the MDCS could stick to video as the output modality for a certain service, but move the video stream from a cell phone to a wall-mounted display (referred to as session mobility [8, 9]). Alternatively, the MDCS could continue to use audio-video as the output modality for another service, but render the audio part on a cell phone and the video part on a TV (non-monolithic rendering [10]).

The MDCS is part of the SPICE platform and hides the complexity and particularities of the different DCS configurations. The advantage of this approach is that different services can use the MDCS and that service providers do not have to implement and maintain their own MDCSs. In addition, service providers will be able to deliver their services to users rather than to the individual devices in the DCSs of these users. This will likely speed up the development and deployment of new services and will probably also reduce their development and deployment costs. This is particularly important as the number of user increase and their DCSs become more heterogeneous and change more frequently. Service providers do however need to be able to inform the MDCS of certain limitations regarding the way in which users may interact with their services. For example, a service provider that delivers real-time video content (e.g., a TV broadcast) may want to guard the quality at which the MDCS delivers this content. It could for instance accomplish this by informing the MDCS that it should not deliver the content to devices with that have very small displays.

To focus our work, we are developing an MDCS for the multimodal delivery of multimedia presentations, which is one specific class of end-user services that will run on the SPICE platform. Apart from traditional distribution and consumption of media, the MDCS is also intended for user-generated content. Because of the benefits of a declarative language for expressing the temporal and spatial synchronisation relationships of a multimedia presentation [11], we use W3C's SMIL language [12].

D. *Terminal Manager*

The Terminal Manager is based on two components, the Terminal Synchronisation and Terminal Configuration.

The Terminal Synchronisation component allows users to have their personal data automatically synchronised with the platform and among different devices. Synchronisation can be summarised as a process ensuring the coherence between two data collection that can be modified separately. A well known example of personal data synchronisation is the Address Book Synchronisation of a mobile phone using SyncML [13].

Since the DCS and in particular network conditions may vary in mobile environments (bandwidth variation, blackout), the synchronisation process is made context and user preferences dependant in order to optimise its task.

The Terminal Configuration component allows the user to install new services on the terminal and to configure installed software and services.

IV. DCS - Use Cases

This section presents use cases illustrating the inter-working of the DCS Management System components. These use cases were extracted from official SPICE project scenarios.

A. *Context aware terminal data synchronisation use cases*

The following message sequence diagram illustrates the exchanges between the Resource Discovery System, Terminal Manager and Dynamic desktop when triggering automatically user data synchronisation upon detection of specific DCS conditions (such as high bandwidth or low cost network detected).

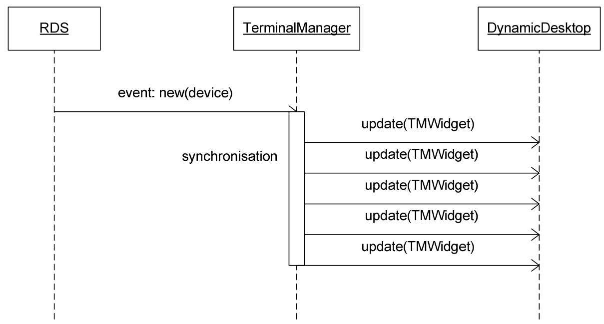

Fig. 2 Context-aware terminal data synchronisation

In this message exchange, the RDS discovers the network change and notifies registered parties of this change, including the Terminal Manager component. The latter evaluate user rules with this new condition and decide to trigger synchronisation. During the synchronisation process, the Terminal Manager sends regular status update to the Dynamic Desktop, which are shown to the user in the Terminal Manager widget.

B. *Multimodal service delivery use cases*

The following section presents two phases of the multimodal adaptation of a media service output to the user.

The first figure presents the initiation phase of a media service, when the user requests the delivery of multimedia content from the service, through this service's Dynamic Desktop widget.

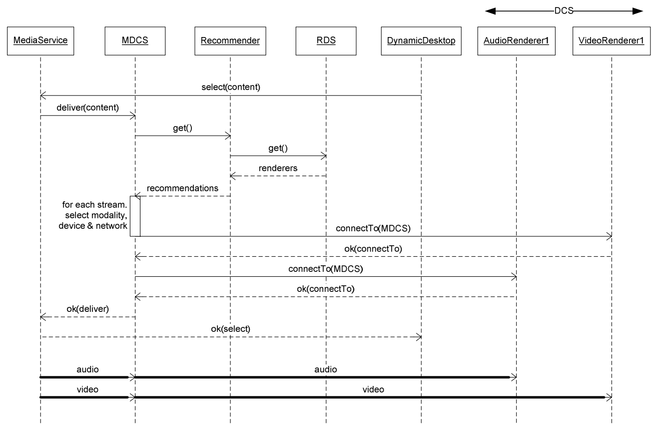

Fig. 3. Multimodal service delivery establishment phase

When the user selects the media item to be delivered in the service widget, this selection is communicated to the media service itself. The service then requests the multimedia delivery from the MDCS, which asks for a recommendation from the Recommender. This component is a KMF component that evaluates user rules against the DCS state to issue recommendations on which device, modality and network to use for the service delivery. Based on the received recommendation, the MDCS decides how the service output should be rendered (e.g. on which device, using which modality, so possibly using a content transformation, etc.) and contacts the concerned renderers.
The second figure illustrates how the MDCS re-evaluates the way the media service content is delivered when a change in the user's DCS is detected.

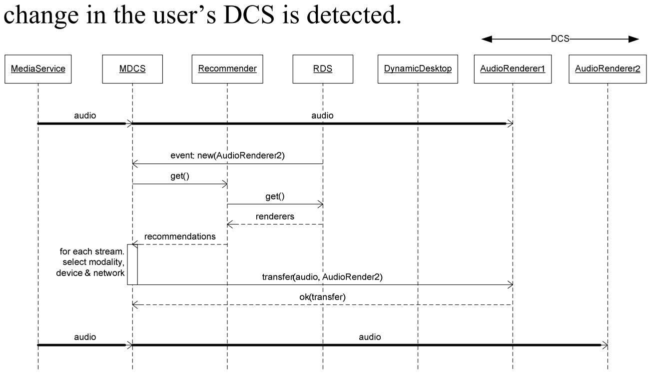

Fig. 4. Multimodal service delivery re-establishment phase

Whenever a change occurs in the DCS, e.g. the appearance of a new device, components that subscribed to this change, such as the MDCS, are notified. It then triggers a new decision process from the MDCS for each running multimedia session.

V. Conclusion

The paper identified limitations imposed by the current service access means in the cases of user-service interaction in multi-device environments, user data synchronisation and service updates. Challenges to overcome the limitations have been summarised and the DCS management architecture has been introduced to address them. However, the paper shows the current progress of the work, indicating how components interact with each other in order to fulfil the most important use cases. Future work includes the implementation of the framework and the research on the detailed aspects as defined by the challenges taking into account current state of the art developments.

Acknowledgment

The authors would like to thank in particular, Rinaldo Nani, Olivier Le Berre, Klaus Moessner, Stefan Gessler, Francois Carrez, Antonio Vitale, Ying Du, Stefan Meissner, Carlo Licciardi, Martin Strohbach and all members of the SPICE project for their contributions.

References

[1] M. Satyanarayanan, "Pervasive computing: vision and challenges," *IEEE Personal Communications*, vol. 8, pp. 10 - 17, 2001.

[2] M. Weiser, "The computer for the twenty-first century," *Scientific American*, vol. 265, pp. 94 - 104, 1991.

[3] S. Oviatt, "Multimodal Interfaces," in *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*, J. A. Jacko and A. Sears, Eds., 2003.

[4] S. Oviatt, R. Coulston, and R. Lunsford, "When Do We Interact Multimodally? Cognitive Load and Multimodal Communication Patterns," presented at the Sixth International Conference on Multimodal Interfaces (ICMI 2004), Pennsylvenia, USA, 2004.

[5] R. Kernchen, P. P. Boda, K. Moessner, B. Mrohs, M. Boussard, and G. Giuliani, "Multimodal user interfaces for context-aware mobile applications," in *The 16th Annual IEEE International Symposium on Personal Indoor and Mobile Radio Communications*. Berlin, Germany, 2005.

[6] M. Sutterer and et al., "Ontology Definition of User Profiles, Knowledge Information and Services," EU IST SPICE Deliverable 4.1, 2007, (checked.

[7] C. Hesselman, H. Eertink, and M. Wibbels, "Privacy-aware Context Discovery for Next Generation Mobile Services," in *The 3rd SAINT2007 Workshop on Next Generation Service Platforms for Future Mobile Systems (SPMS 2007)*. Hiroshima, Japan, 2007.

[8] W. Schulzrinne and E. Wedlund, "Application-Layer Mobility using SIP," *Mobile Computing and Communications Review (MC2R)*, vol. 4, pp. 47 - 57, 2000.

[9] R. Shacham, H. Schulzrinne, S. Thakolsri, and W. Kellerer, "Session Initiation Protocol (SIP) Session Mobility draft-shacham-sipping-session-mobility-03," IETF November 13, 2006,http://www.ietf.org/internet-drafts/draft-shacham-sipping-session-mobility-03.txt, (checked 31th January 2007).

[10] P. Cesar, Bulterman, and Obranovic, "Non-Intrusive User Interfaces for Interactive Digital Television Experiences.," in *European Conference on Interactive Television*, 2007.

[11] P. Cesar, D. C. A. Bulterman, and A. J. Jansen, "An Architecture for End-User TV Content Enrichment," *Journal of Virtual Reality and Broadcasting (JVRB)*, vol. 3, 2006.

[12] D. C. A. Bulterman and L. Rutledge, *SMIL 2.0: Interactive Multimedia for Web and Mobile Devices*. Heidelberg, Germany: Springer, 2004.

[13] Open Mobile Alliance (OMA) Ltd., "SyncML Data Sync Protocol," OMA-SyncML-DataSyncProtocol–V1_1_2-20030612-A, 12th November, 2003, (checked.