Position Papers of the 2013 Federated Conference on
Computer Science and Information Systems pp. 45–50

USSD communication channel as alternative to XML SOAP in mobile Unified Communication applications

Dariusz Bogusz
Siemens Enterprise Communications*
ul. Mińska 63A
03-828 Warsaw, Poland
Email: dariusz.bogusz@siemens-enterprise.com

Grzegorz Siewruk
Warsaw University of Technology
Faculty of Electronics and Information Technology
ul. Nowowiejska 15/19
00-665 Warsaw, Poland
Email:gsiewruk@gmail.com

Jarosław Legierski
Orange Labs
ul. Obrzeżna 7,
02-691 Warsaw, Poland
Email: jaroslaw.legierski@orange.com

Jan S. Kunicki
University of Warmia and Mazury
Faculty of Mathematics and Computer Science
Słoneczna 54 Street
10-710 Olsztyn
Email: jan.kunicki@matman.uwm.edu.pl

Abstract—**Traditional mobile Unified Communications (UC) clients use data channel over GSM/UMTS network for communication with UC services broker and providers. UC signaling traffic characteristics require a large number of the short data connection sessions and can cause overload of the mobile network. The reason is a ratio of the overhead data to the useful signaling information.**

This paper presents an analysis of the mobile UC architectures and a possible use of Unstructured Supplementary Service Data (USSD) channel for UC application in the mobile domain. The usage of USSD channel can help to provide more effective applications for the mobile UC user.

I. Introduction

UNIFIED Communications is a new trend on the enterprise market. The basic idea is to handle all communications of a business user (voice call, video, instant messaging, teleconferencing, e-mails, voice mails, web collaboration, mobility and customer contact center etc.) over a single scalable, software-based, unifying communications platform working with any IT, voice, and application environment. Some of the key features are:

- one universal telephone number independent on a physical device (desk phone, soft phone, home office, mobile),
- presence management - including personal and media status,
- mobility - including virtualization of the communication services irrespective of a client (softphone on PC, browser based client, personal voice based client, mobile smartphone client),
- communication enabled business processes CEBP – multimedia communication built into the enterprise business applications via API or the available plugins.
- social media integration – possibility to communicate via Web 2.0 portals in the Cloud via API or federation

Architecture of a typical UC system is presented in Fig. 1.

Proliferation of the smart mobile terminals allowed to offer UC users a dedicated mobile application that significantly increases functionality of the office communication systems for the business end users. Integration of the UC applications with other systems [1], [2] allows to offer a complete end extensible environment. In the next sections we discuss evolution of the architectural approach to the mobile UC, we also suggest a possible optimization in this area and present a prototype implementation.

II. Mobile UC Applications Evolution

Mobility of the end users is a very important aspect for many enterprises. This section presents evolution of the enterprise mobile applications architectures based on the solutions available on the market during the last 20 years.

A. *Hardware DECT/GSM system*

The first mobile application was announced in 1995 as a joint trial development project between TELIASONERA AB; ERICSSON Inc. [6]. Ericsson was to deliver

5,000 dual-mode GSM/DECT handsets, which should operate over both wireless networks. This concept was beneficial to the telecommunications providers due to better utilization of RF, and to the customers because of better radio coverage and maintaining in-campus mobility.

However such a system requires a specialized terminal (hardware) capable of using DECT and GSM transmission (GSM means here GSM900 as well as DCS1800) [7]. Such terminal must be able to maintain two concurrent communication sessions; one over GSM and the other over DECT. Additionally it should be available over one personal number irrespective of the terminal location; via DECT in campus and via GSM outside. Intelligence is located in the network infrastructure, no intelligence in the terminal.

With all these advantages this model requires a specialized dual-mode GSM/DECT handset. This in turn increases price of the whole solution and diminishes battery life. Some vendors tried to use another approach.

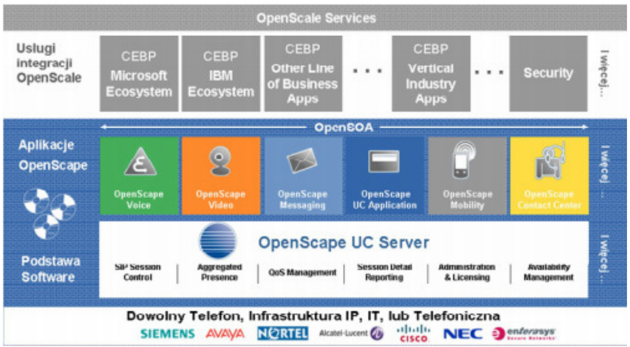

Fig. 1 OpenScape UC system architecture

B. Corporate GSM

The idea of corporate GSM has been presented by many vendors. The first fully developed architecture named DWOS, was presented by Ericsson[8]. Implementation of the critical functionality is split between the enterprise and telecommunications provider core infrastructure [14]. This unfortunately requires many integrations between the provider and enterprise telecommunication in both organisations and in the infrastructure. The market has not developed any appropriate business model and such solutions have not been widely adopted.

C. OpenScape UC Mobile Client

By the year 2002 the first Unified Communications systems were announced to the market [12]. They promised many advanced features irrespective of the user location, administrative domain, or access network technology. One of the first UC systems in the market was Openscape UC issued by 2004 [13]. In this paper we cover the mobility aspect of this solution. Openscape UC mobile Client is mainly CTI (Computer Telephony Integration) client extensively using WiFi or GSM Data channel for UC signaling traffic. The payload (voice only) to the mobile client is transmitted over a standard GSM channel. GUI of this client is a simple implementation of UI for managing Web Services responsible for the advanced UC features. Openscape UC mobile Client has no built in intelligence, which could allow us to name it an agent system.

D. Mobile Connect

At CEBIT 2006 Siemens presented a prototype, and at CEBIT 2007 the commercial product of a different Architecture, commercially named HiPath Mobile Connect. Mobile Connect is a solution based on a dual connectivity (WiFi and GSM) of the commercially available mobile phones [9].

The main architectural difference is the application server called FMC Appliance, located within the enterprise's IP network. It is fully managed by the Enterprise. As the contractual relationships do not go beyond a standard commercial contract between the enterprise and telecommunications provider, such system is commercially much easier to deploy. Thus much wider adoption of this model. With a lack of information from the mobile provider, a network implementation of the advanced services like seamless handover between enterprise WiFi and public GSM is implemented partially as FMC client function. It means that more intelligence is moved to the endpoint. This architecture implements for the first time FMC client as an intelligent agent system interworking with FMC Controller through implementation of its advanced functions.

E. OpenScape MObile

Openscape OSMO (OpenScape MObile) is a client combining rich UC functionality of OpenScape UC Mobile Client with SIP client implementation on a smartphone [15]. Though such functionality is beneficial for the users it poses many challenges to the network.

OSMO client maintains two parallel connections:

- CTI connection to a UC proxy named OpenScape Facade
- Payload and SIP Signalling connection to an enterprise Session Border Controller (SBC) named OpenScape SBC

This architecture is necessary (despite its complications), because SIP is a standard protocol. Standard voice and video functions are implemented within SIP signalling accessing the enterprise net via SBC. All non standard UC protocols are transmitted via data channel to a UC proxy. Such architecture with split of functions requires the client on a smartphone to perform as an agent. User interface of UC mobile client OSMO is presented in Fig. 2.

Fig. 2 OpenScape OSMO GUI on iPhone

III. Mobile UC Signaling traffic

A. Data traffic from UC mobile application

Signalling traffic characteristics of such an agent in mobile domain creates an interesting problem with many practical implications for the mobile provider in the signalling and data path. They are also important from the enterprise security and availability perspective. Issues resulting from integration of those two worlds are planned for further investigation. Here we cover only some aspects of it.

Typical Unified Communication mobile user uses the application installed in his mobile phone to:

- Change personal status (available, free, out of the office on holiday, etc.)
- Change preferred device (office phone, mobile phone , home phone, etc.)
- Read phone log (answered/ not answered calls, connections redirected to another device, etc.)
- Read telephone book records (global and personal address books)

The most frequently used function focuses on the change of personal status of a user. The end users can change their status using OpenScape Mobile application. Request of the status change is transmitted to Unified Communication server as Web Service call using XML data and SOAP [3] protocol. The connection is established as data channel via IP network. Using this method every status change results in transmission of 4890 bytes in 4 TCP packets (without TCP/IP packet headers).

Information of Unified Communication subscriber status change can be transmitted using 3-5 digits USSD code. Fig. 4 presents USSD communication captured using Wireshark in SIGTRAN (SS7 over IP protocol) MAP messages. The amount of data sent using USSD (without packets headers) is about 632 bytes in SCTP protocol in both directions (between USSD gateway – server exposed API).

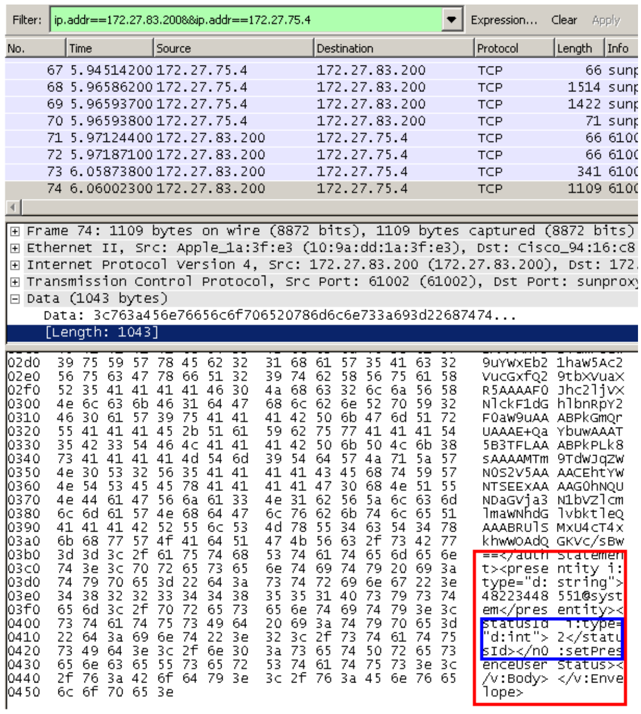

Fig. 3 TCP packet with user status change

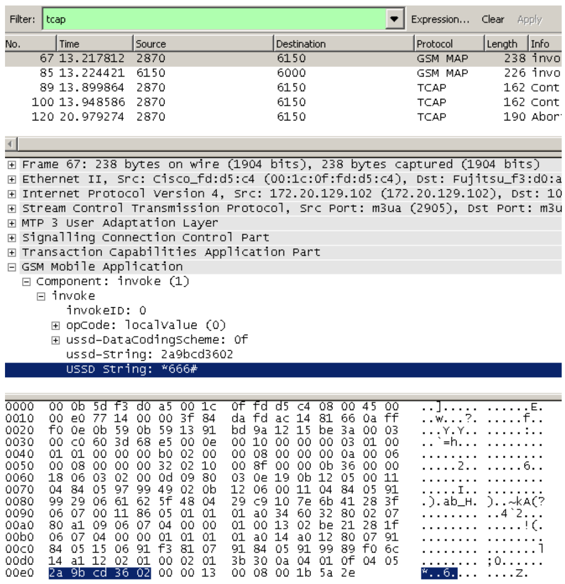

Fig. 4 TCP packet with USSD code

Fig 5 presents typical usage of OSMO application based on the application logs recorded at a mobile phone.

From the data presented at Fig 5 we notice that the number of communication requests per hour per user ranges between 1 and 21. Average value is more than 6 (6,42). Technically UC system can grow up to 100 000[13] users. By a conservative assumption of 1000 users/enterprise we come to 6,42*1000=6420 requests per hour. The observed application traffic characteristics re-

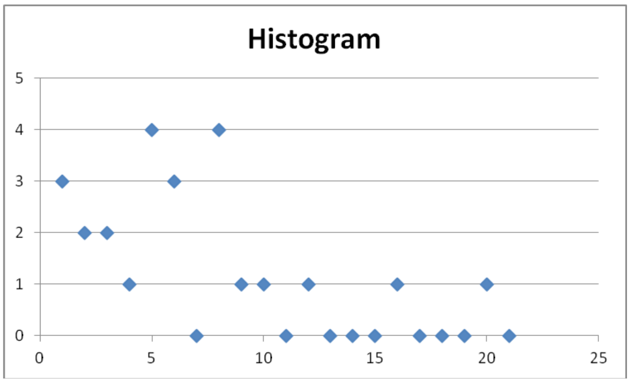

Fig. 5 Histogram of personal status change requests in time for OSMO UC application. x-req/h, y-frequency over 48h

sults in a large number of short data connections over the mobile network (GSM or UMTS). The number of valid data (one Integer value) transferred in TCP/IP packets (Fig. 3) is very small in comparison to the amount of other data (TCP/IP packet headers, XML headers etc.).

We have also expected this data to exhibit more distinctively other structures like periodic and non-periodic components, etc., important for better understanding of this traffic nature. These however did not show with the data presented at Fig. 5. Such intriguing issues are to be subject to our further investigations.

B. USSD communication channel

Based on the information presented in section II, we can conclude that using the data channel for communication Mobile Application – UC system is not an optimal solution. The reason is the amount of the signalling data in relation to the payload data and the large number of a short time data connections (Fig. 5). Our concept focuses on using the USSD channel (rather than data-over-Internet) for communication between the mobile application and the UC server.

Unstructured Supplementary Service Data (USSD) is a communication protocol implemented in the popular mobile networks (GSM, UMTS). The main role of USSD is to provide the two sided communication between a mobile phone and various applications and the system on the telecommunications provider side.

The first USSD specification (Phase 1), described in GSM 02.90 documentation [10] supports only the communication initiated from a mobile phone – pull mode (using dedicated USSD codes e.g. *665*11#, *100*# etc.) The Second Phase specified in GSM 03.90 [11] supports both communication methods originated by the network (push) and terminal (pull). The network initiated communications allow sending information to the mobile terminal.

From a functionality point of view the USSD message contains up to 182 characters. Compared to Short Message Service (SMS), USSD is a real-time transmission using the signalling channel of the mobile network.

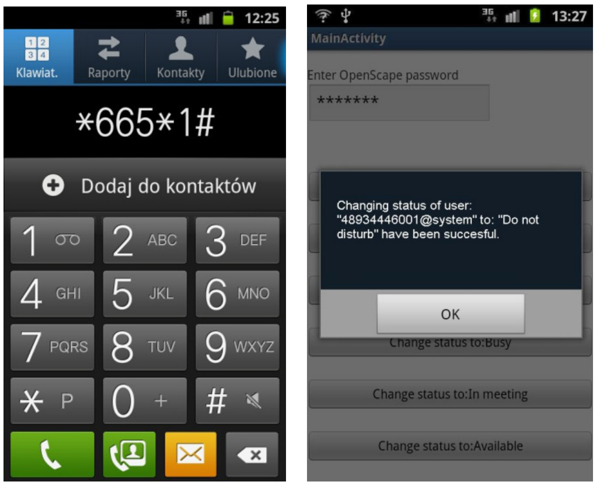

Fig. 6 Mobile phone originated USSD (left) and network originated USSD message (right)

In the mobile provider core network, USSD communication is implemented using MAP (Mobile Application Part) protocol. MAP is an application layer protocol dedicated for the mobile networks based on SS7 signalling stack. The USSD communication is supported using the following MAP messages:

- MAP_PROCESS_UNSTRUCTURED_SS_REQUEST – pull operation between terminal (originator) and application server,
- MAP_UNSTRUCTURED_SS_REQUEST- push USSD messages,
- MAP_UNSTRUCTURED_SS_NOTIFY – USSD - push operation.

USSD offers session-oriented connections and this functionality allows to develop applications supporting dialogue with the end user (e.g. using interactive menu displayed on a mobile terminal).

IV. SYSTEM ARCHITECTURE

The solution presented in this paper was implemented as a low cost application for UC mobile end user. Therefore, we expect it to be highly effective. The high level system architecture is presented at Fig. 7. In order to create a scalable service in line with the latest trend in the field of mobile services, our architecture consists of:

- OpenScape UC server – actual UC services provider
- Web server, which processes USSD requests to UC API
- Client - on user smartphone for operations like: login or changing users presence status

This section presents the implementation and configuration of the client endpoint and software architecture of USSDforUC environment.

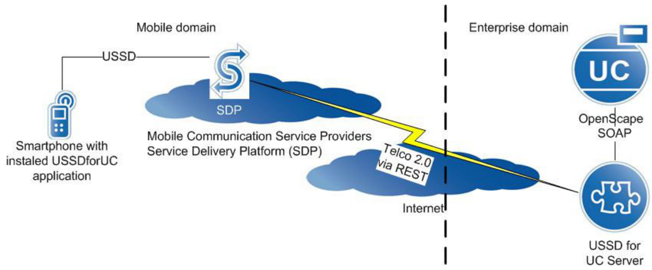

Fig. 7 USSDforUC High level system architecture

As presented at Fig. 8, *USSD* message sent by the user is redirected via provider's API to the USSDforUC application server which communicates using SOAP Web Services API with Open Scape UC server. Both communications: USSD and communication with UC server are created within the request-response model using Web Services. Because of the session timeout implemented in USSD, the USSDforUC server must send response to the particular request during the same communication session and in a limited time. The USSDforUC was developed as an application without its own database and the information about UC users and their settings (passwords, telephone numbers) is stored in Openscape UC repository.

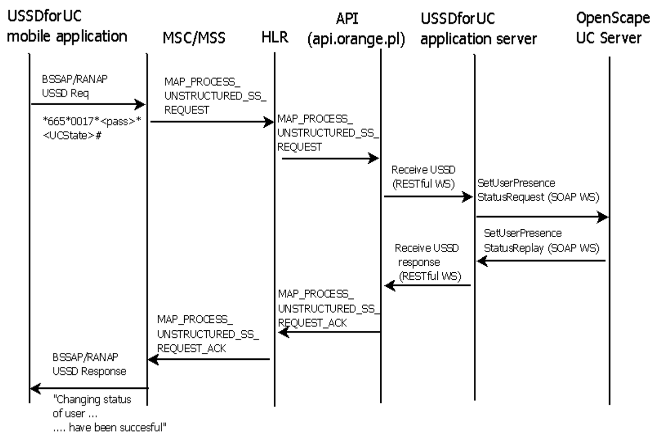

Fig. 8 APIs and telecommunications protocols request/response flow. MSC – Mobile Switching Center, MSS – Mobile Softswitch Solution, HLR – Home Location Register

A. USSDforUC server

This section describes configuration and implementation of USSDforUC server which translates USSD message from expressed in Internet provider's APIs (based on Restful Web Services [4]) to SOAP messages. The main objectives executed by this module are:

- login to OpenScape UC server using UC API,
- creation object represents OpenScape user profile in the memory,
- changing OpenScape user presence status to the one of the list: Do not disturb, Be right back, Unavailable, Busy, In meeting, Available.

User profiles are stored in the memory in a way which does not overload the system. Only the necessary data like login name and mobile phone number are saved in the profile. In order to maintain user base correctness the array of users is refreshed every 2 hours.

Changing the user presence status is more complex. The architecture of OpenScape UC server does not allow to use the technical account dedicated for API calls to change the particular user presence status. The only solution is to login via API as the user who wants to change his or her status and to complete the task from that level. Information about the user login and password is coded in USSD request as a special code. UC user account is recognized using MSISDN number from USSD request. Server looks for MSISDN in the array of profiles stored in its memory in order to login to the user account and change status.

B. USSD mobile application

This application has been developed as a mobile application for Android Operating System.

Application itself is created as a single activity (android screen) which contains a single text field and six buttons. To complete the first request it is mandatory to enter password before clicking any button. Otherwise the user will be provided with an error message. The USSDforUC application is actually using only one Android SDK function: 'MakeCall', with one parameter i.e. telephone number. API call requires the phone to set up connection with the number it gets. In our case the number is the USSD request in form of *665*17*code*, where:

- 665 is the USSD server id,
- 17 is application id (USSD server is aware of the URL it should redirect USSD messages to, if they come with particular id),
- code which contain the mash up of password and status code.

C. Security aspects

As is mentioned in section *B*, the message which contains the user password is sent through an unsecured USSD channel. Because of a simple USSD coding in SIGTRAN messages (Fig. 8) use of any network sniffer enables to read the passwords. To increase the security, we have used a simple encryption method. Every special character sign of the code page is replaced by a number. This method is not strong one, but this addition results in a random generated sequence of alphabet letters and is hard to decode for a person not knowing the key. To harden against a dictionary attack (trying to login on account with each combination of letters) we have implemented a module which counts the unsuccessful login attempts, adds them to server log and disables a user account for 1h hour.

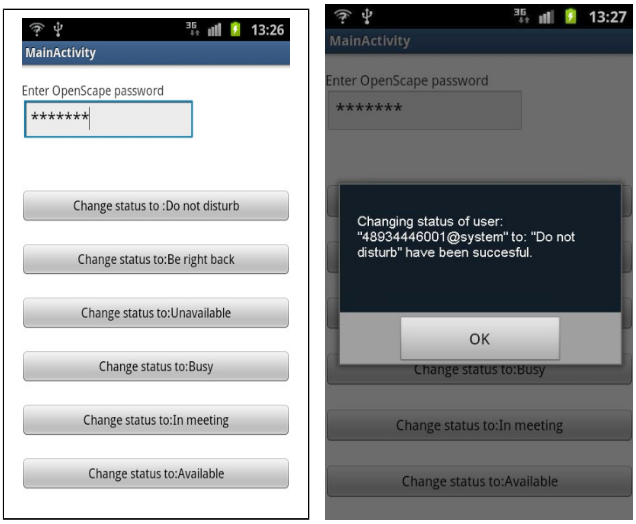

Fig. 9 USSDforUC mobile application GUI on Android

V. Summary

This paper presents the traffic characteristics of a modern mobile UC user. It discusses the possible drawbacks of transmitting UC signaling traffic over IP in the mobile domain as well as the possible use of USSD channel in UC mobile application including a prototype implementation.

USSD communication allows bidirectional, half-duplex connections between the mobile terminal and application. This communication is completely transparent for the handset and mobile network. USSD can be used by any type of a mobile phone and does not depend on the vendor and operating system.

Using USSD channel for signaling, mobile UC application can avoid the large number of short time data connections. Because in the core provider's network USSD messages are coded using binary protocol the single UC status change request is transmitted using about 632 bytes as compared to 4890 bytes transmitted in the current solution - SOAP and XML based Web Services call in data traffic.

An interesting feature of the presented USSDforUC prototype is that the implemented system doesn't use any database what simplifies its overall system architecture.

Using USSD for UC signaling traffic shows that even the leading commercially available UC solutions have areas where they can be vastly improved.

Characteristics of UC signaling traffic in the mobile domain as well as on the interface to other ICT systems in the enterprise, telecommunications provider as well as in the public cloud will be subject of further investigation. The research area is to model and manage this traffic to avoid communication bottlenecks and to manage user experience.

Acknowledgment

Prototype of USSDforUC service was developed under the program Orange Labs Open Middleware 2.0 Community [5].

*) Siemens Enterprise Communications Sp. z o.o. is a Trademark Licensee of Siemens AG

References

[1] D.Bogusz, J.Legierski, A. Podziewski, K. Litwiniuk, „Telco 2.0 for UC – an example of integration telecommunications service provider's SDP with enterprise UC system", Wrocław 2012. Results of operation in 2010, Warsaw 2011

[2] D. Bogusz, P. Korbel, J. Legierski, Integracja systemów Unified Communications z platformami usługowymi operatorów, KSTiT 2011 conferencing materials, Przegląd Telekomunikacyjny 8-9/2011

[3] SOAP specification W3C, http://www.w3.org/TR/soap/ [5.03.2013]

[4] L. Richardson, Sam Ruby, David Heinemeier Hansson, RESTful Web Services, O'Reilly, 2007

[5] Open Middleware 2.0 Community portal – http://www.openmiddleware.pl

[6] "Telia, Ericsson plan dual-mode, GSM/DECT", Electronic News (10616624);4/17/95, Vol. 41 Issue 2061, p28, April 1995

[7] Patric Lind, "Terminal aspects of DECT/GSM as an evolution towards UMTS", http://cordis.europa.eu/infowin/acts/analysys/products/thematic/mobile/ev_cellular_sys/dect_gsm.html

[8] Rune Johanson, Mikael Nilsson and Torbjorn Ward,"Mobile Advantage Wireless Office—A digital wireless office system for TDMA/136 networks", Ericsson Review No. 1,1999 19

[9] "Demystifying Enterprise Fixed Mobile Convergence", A white paper from Siemens Enterprise Communications, February 2007

[10] Unstructured Supplementary Service Data (USSD); Stage 1 3GPP specification: 02.90 www.3gpp.org/ftp/Specs/html-info/0290.htm

[11] Unstructured Supplementary Service Data (USSD); Stage 2 3GPP specification: 03.90 http://www.3gpp.org/ftp/Specs/html-info/0390.htm

[12] "Siemens ICN Announces LifeWorks Concept for Integrating Personal, Enterprise and Carrier Technologies", http://www.prnewswire.com/news-releases-test/siemens-icn-announces-lifeworks-concept-for-integrating-personal-enterprise-and-carrier-technologies-71170057.html

[13] "Siemens Redefines Communication Over IP Market With Announcement of HiPath(TM) 8000 Real-Time IP System for Very Large Enterprises", http://www.prnewswire.com/news-releases/siemens-redefines-communication-over-ip-market-with-announcement-of-hipathtm-8000-real-time-ip-system-for-very-large-enterprises-72780072.html

[14] "HiPath CorporateConnect V2.0, CorporateConnect Server", Service Guide A31003-G3620-S300-4-7620, by Siemens AG 04/2006

[15] "OpenScape Mobile V7R1, Administrator Documentation", A31003-M4070-M100-8-76A9, by Siemens Enterprise Communications, 12/2012