

A Service Creation Environment for interactive, menu-driven Mobile Services

Jürgen Tacken, Thorben Janssen, Stephan Flake, Dirk Fischer
ORGA Systems enabling services GmbH, 33104 Paderborn, Germany
{jtacken, tjanssen, sflake, dfischer}@orga-systems.com

Abstract

In this article, we describe a Service Creation Environment (SCE) that allows for protocol- and technology-independent development, deployment, and maintenance of interactive, menu-driven mobile services for cellular networks. The presented SCE eases the maintenance of mobile services, in particular by means of versioning and access right management. The SCE even allows third-party content providers to deploy services on their own by remotely accessing the SCE residing at a Mobile Network Operator (MNO). The presented SCE is based on an open source Content Management System and currently supports maintenance of HTML-based service descriptions.

1. Introduction

In addition to making plain voice phone calls, MNOs nowadays offer their subscribers a variety of non-voice mobile value-added services (VASs), e.g., messaging services like SMS or data services like GPRS. Popular VAS applications in this context are messaging applications like stock news or sports news via SMS or data services like downloads of games, pictures, audio and video content via GPRS. These VAS applications (or: end-user services) are typically implemented and deployed using different VAS standards and technologies within a cellular network. Nowadays, there is only low or even no tool support for the development, deployment, and maintenance of such multi-technology VAS applications. Most tools are only applicable to develop, deploy, and maintain VAS applications based on a single technology, e.g., IN Switch offers a number SCEs, each for a different distribution/transport technology [1]. Therefore, it is currently a very hard and time-consuming task to maintain all the different available end-user services in the network of an MNO. Service developers need to be experts for different technologies and have to be able to work with many different tools. A single tool to

support the development, deployment, and maintenance of all the different end-user services would ease their work drastically and also save time and money.

This article describes a SCE that is a big step towards such a single tool for developing, deploying, and maintaining end-user services while supporting different VAS standards and technologies. The main goal is to develop a protocol- and technology-independent solution. There are already some SCEs (e.g., TribalText Studio [2]) but currently none of these SCEs fulfills all of our requirements. The SCE must support the development, deployment, and maintenance of all end-user services for different mobile non-voice messaging and data VAS. The integration into the current network infrastructure of the MNO should be easy. Other SCEs like the TribalText Studio require other proprietary components like their interactive messaging engine, billing engine, and content delivery system.

The paper is structured as follows. In Section 2, relevant base technologies for our SCE are described. In Section 3, fundamental infrastructure elements and services of GSM networks are explained. The features, implementation, and usage of the proposed SCE are presented in Section 4. Basically, our SCE is based on an open source Content Management System (CMS), and as a common format for the design and development of end-user services, the SCE uses HTML. Concluding remarks and an outlook are given in Section 5.

2. Base Technologies

On the one hand, we review techniques to present subscribers the VASs and end-user services available in a cellular network, focusing on means to *interactively navigate* through the available services. On the other hand, we have to look at techniques that support the life-cycle management of services, applications, or – more generally – digital content.

MNOs offer their subscribers a variety of VASs that can be considered as transport channels and enablers for end-user services. The kinds of end-user services differ from country to country and types of subscribers

(prepaid/postpaid), but the currently most widely deployed and used end-user services are (a) messaging services with pictures and videos, (b) information services (e.g., stock news, weather forecasts, and other kinds of news – either textual or voice-based), as well as (c) downloads of ring tones, games, music, etc. Appropriate navigation facilities (i.e., menus) have to be provided to allow MNO subscribers navigating through the set of offered VAS and end-user services with their handsets. The different end-user services are typically ordered in a hierarchical tree-like manner. In this context, SIM Toolkit is currently widely employed [3]. SIM Toolkit is an ETSI standard for SIM cards and GSM phones allowing to load applications (or: end-user services) into SIM cards with the possibility to update them via SMSs. To update a SIM Toolkit application selection menus can be inserted/alterd/removed by the MNO. Note that an update usually requires to send multiple SMS to all deployed SIM cards. In this sense, SIM Toolkit is rather static and one-way, such that it is less suitable for flexible service structures that might change frequently. In contrast, we aim at a more server-oriented navigation approach based on *wireless Information Query* (wIQ) [4] that avoids such an issuer-centric and broadcast updating technique (see Section 3).

CMSs are dedicated tools to (remotely) manage the life-cycle of different kinds of digital content. There is already a vast number of commercial as well as open source CMSs.¹ Though most CMSs are dedicated to managing static and dynamic Web pages, they can also be employed for our purposes, i.e., to support the deployment of new VAS applications within the service platform of an MNO. Based on the main system requirements listed in Section 1, we evaluated some exemplary suitable open source CMSs, i.e., the PHP4.0-based Contenido² and PostNuke³ as well as the J2EE-based Nukes on JBoss [5]. Generally, most CMSs nowadays provide a convenient GUI and support user (or: access right) management – though with different granularity. For our desired SCE, however, we consider it more important to have support for content versioning and system extensibility. It turns out that Nukes on JBoss offers the best support for system extensibility due to its object-oriented modular structure; new system modules can be easily developed and integrated by means of inheritance from according base classes. Table 1 summarizes our evaluation results w.r.t. the main requirements and differences. As a result, we choose Nukes on JBoss as a basis for our SCE. For details of these evaluation see [6].

¹ For an overview, see <http://www.cmsreview.com>

² <http://www.contenido.org>

³ <http://www.postnuke.com>

Table 1: CMS Evaluation

	Contenido	PostNuke	Nukes on JBoss
User Management	++	++	+
Content Versioning	--	--	++
System Extensibility	+	+	++

++: supported functionality --: unsupported functionality

+: supported functionality with minor restrictions

3. Service Infrastructure

The underlying infrastructure for the work described in this article is the GSM network. GSM (Global System for Mobile Communications) is a world-standard for mobile telecommunication. We here only give a short overview of the VASs in a GSM network that are relevant for our work. For more details, we refer to [7].

Mobile non-voice VAS in a GSM network, are – among others – Short Message Service (SMS), Unstructured Supplementary Services Data (USSD), Wireless Application Protocol (WAP), and General Packet Radio Service (GPRS). These services will be explained in more detail in the following subsections. We also briefly present ORGA Systems' wireless Information Query platform (wIQ), which enables MNOs to provide VASs to their subscribers.

SMS is a well-known messaging service in GSM networks. It enables subscribers to send and receive text messages to and from mobile phones. A single short message can have up to 160 characters of text when Latin alphabets are used (70 characters when non-Latin alphabets such as Arabic are used). Non-text based short messages (for example, in binary format) are also supported. It is possible to send messages with more than 160 characters with several “chained” SMSs.

SMS is a store and forward service, in other words, short messages are not sent directly from a sender to the recipient, but via an SMS Center. Each mobile telephone network that supports SMS has one or more messaging centers to handle and manage short messages.

USSD is a very similar service to SMS. USSD text messages can be up to 182 characters in length. But unlike SMS, USSD is not a store and forward service. It is session-oriented, such that when a subscriber accesses a USSD service, a session is established and the radio connection stays open until the subscriber, the application, or a time-out releases it. This means that the turnaround response time for interactive applications is shorter for USSD than SMS.

To use an USSD service, the subscriber has to enter a combination of numbers with ‘*’ and ‘#’ characters.

Examples for valid USSD calls are “*100#” or “*113*3#”. This means that a subscriber does not need to access any particular phone menu. He can directly enter USSD commands. Therefore USSD works on all existing GSM mobile phones.

WAP is an open international standard to access the Internet from a mobile phone. WAP allows to use mobile services in a way known from using a Web browser. It thereby combines two of the fastest growing industries: mobile networks and the Internet. Nevertheless, WAP is simply a protocol – a standardized way mobile phones talk to a server installed in the mobile phone network.

GPRS is a data service that allows information to be sent and received across a mobile telephone network. It is a packet switched protocol. GPRS facilitates "connectionless connections", also referred to as “virtual connections”, in which the subscriber is not always transmitting information but is ready to do so in an instant as the need arises. A GSM error-corrected circuit switched data channel supports a data rate of 14.4 kbit/s, whereas GPRS supports a higher data rate up to a theoretical maximum of 140.8 kbit/s, though typical rates are closer to 56 kbit/s.

UMTS as the successor of GSM supports up to 1920 kbit/s data transfer rates, although subscribers in the real networks can currently only expect a performance of up to 384 kbit/s – in Japan upgrades to 3 Mbit/s are in preparation. GSM services like SMS, USSD, and WAP will still be available in UMTS networks.

The **wIQ** platform is an example for a platform that enables GSM and UMTS network operators to provide all of their subscribers with interactive, menu-driven end-user services [4].

The content of these services is based on HTML. All data is hosted on a standard Web server provided, e.g., by a content provider (see Figure 1). Interfaces such as CGI and PHP for creating dynamic HTML pages are supported. For the data sent from and to mobile phones of subscribers, USSD and SMS can be used. The phones do not need to have WAP or SIM Toolkit capabilities. Even mobile phones of the first generation are supported. Therefore, operators can reach every subscriber with services distributed via wIQ.

Since wIQ is a completely server-based solution, applications can be modified without an impact on SIM cards or handsets. Once wIQ is set up in the network, every subscriber can use the system immediately. This significantly reduces the total cost for system maintenance.

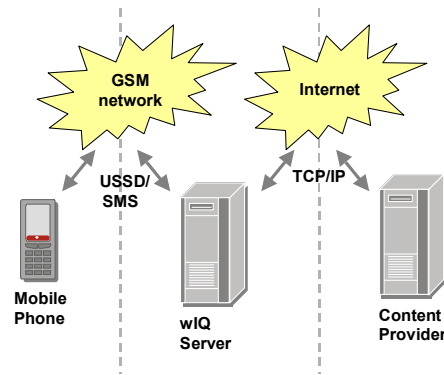


Figure 1: wIQ Architecture

4. Service Creation Environment

Our SCE is realized as an extension of Nukes on JBoss [5], as this CMS provides the greatest benefit of the considered CMSs (cf. Section 2). Like all other investigated CMSs, Nukes on JBoss provides a remotely accessible graphical user interface (GUI) and an access right management system. The main advantages of Nukes on JBoss are the underlying JBoss application server and an integrated version control system. Especially the JBoss application server and the use of J2EE technology ensure convenient system expandability and aggregation of external systems, which are important features w.r.t. future enhancements of the SCE.

4.1. Features

Besides the SCE’s primary purpose to be able to conveniently export newly created and modified end-user services to productive service provisioning systems, the SCE currently comprises the following main features:

- The user management allows for controlled (remote) access to end-user services and corresponding configuration data for editing, deployment, and maintenance purposes.
- A GUI allows SCE users to create and manage interactive, menu-driven end-user services (e.g., based on wIQ) in an easy and intuitive way without in-depth knowledge of the underlying technology.
- HTML code is validated before it is stored in the database. Minor syntax errors are automatically corrected.
- The SCE provides a version management for developed and deployed end-user services and configurations. This allows fallbacks to previous deployed versions, e.g., in the case of problems.
- To guarantee permanent availability of the end-user services, the SCE allows to install, edit, and test

services and corresponding configurations offline first, i.e., end-user services can be tested without affecting the productive system. This avoids unexpected system failures caused by invalid configuration data.

4.2. Architecture and Functionality

As already mentioned, the SCE is an extension of Nukes on JBoss. The SCE is deployed and executed inside the JBoss Application Server, which provides different services to the SCE (e.g., database connectivity). A MySQL database is used for data persistence. The SCE is accessed via HTTP by using a web browser. Figure 2 shows an overview of the system architecture. The depicted external systems refer to multiple service platforms and content servers, on which contents and the configurations of the managed services are stored.

The functionality of the system can be divided into service management, export, and general functions. All functions can be accessed using the provided GUI.

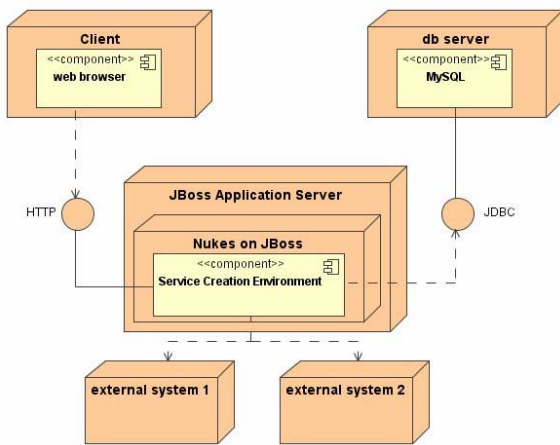


Figure 2: SCE Architecture

Service management comprises all functions that are needed to administer end-user services and configurations. All functions deal with the content and the specific configuration of an end-user service. The functions for (a) managing the service configuration data and (b) actually exporting the services are pooled in the group *Export*. Service export is divided into (b1) export of the service content and (b2) export of the service configuration data, which may be stored on a different server. Finally, the *General* functions are provided by Nukes on JBoss, i.e., Version Control and User Management.

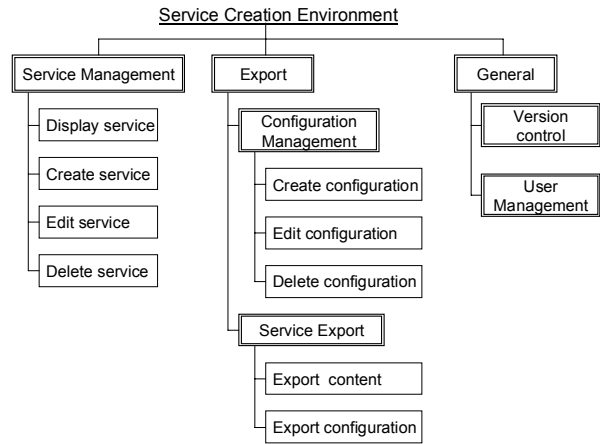


Figure 3: SCE Function Tree

4.3. Application Example

Figure 4 shows a screenshot with the SCE dialog for creating a new end-user service. For each service, a filename and the HTML code has to be specified. The filename is used to identify the service in the SCE and to store it on the productive Web server. It is also used in the wIQ configuration to provide access to the new service. HTML code is validated before it is stored in the database. This prevents a service downtime due to syntax errors.

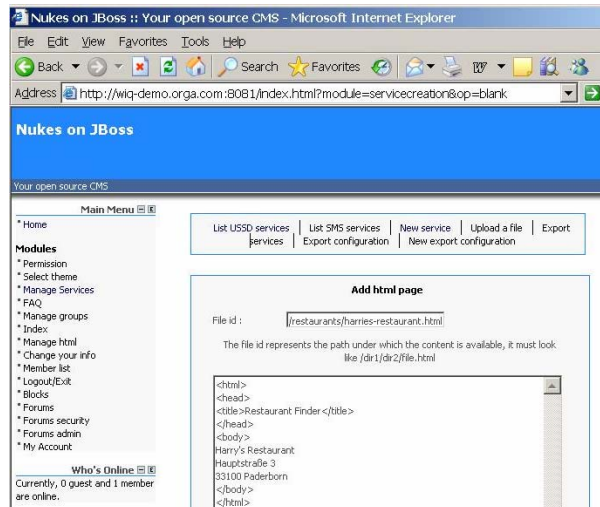


Figure 4: Service Creation

After a new service has been created, it has to be exported to the productive system (e.g., wIQ). If this is done and the service is activated, it can be accessed by subscribers who enter the configured USSD service code into their mobile phones. Figure 5 shows how the end-user service entered in Figure 4 will be displayed to the subscriber.



Figure 5: Display new service

When editing an existing end-user service (see Figure 6), one can choose whether to store the edited content as a new version or to overwrite the existing content.

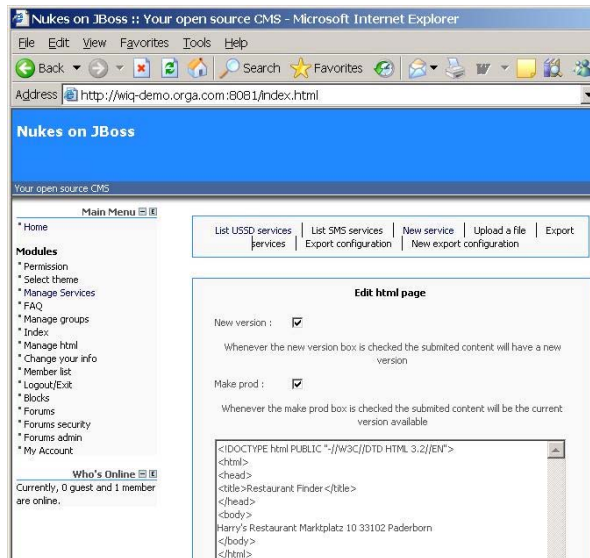


Figure 6: Service Editing

In addition, one can choose whether the new version should be the new productive version or whether the currently productive version should be kept. This allows to edit a service, put it into production, and go back to the previous version if the new one contains an error.



Figure 7: Display Edited Service

After editing the created service as shown in Figure 6 and exporting it again to the productive system, subscribers will see the new content. Figure 7 shows how

the new service content is displayed on a mobile phone.

5. Conclusions & Outlook

We see an SCE as a vital enabling tool to allow content providers to manage end-user services on their own while making use of existing mobile network infrastructures. In this article, we presented our SCE that allows for protocol- and technology-independent development, deployment, and maintenance of interactive, menu-driven mobile services in cellular networks. After a survey and evaluation of different existing CMSs, we chose Nukes on JBoss as a basis for our prototype SCE. Nukes on JBoss, which already met most of our system requirements (GUI, version management, and access control management), has been selected because of its convenient extension capabilities.

As a case study to demonstrate the applicability of the approach, our SCE has been applied to implement and deploy menu-based end-user services in the GSM network.

Although we only made use of USSD and SMS as the underlying mobile VAS, the SCE allows to easily integrate other messaging and data VAS like MMS, GPRS and WAP. Currently, the internal format for the service representation and editing is HTML. But in order to be more flexible and technology-independent, we are currently evaluating an appropriate XML representation of VAS applications.

Another future research topic is to be able to easily integrate context information into the service creation process, e.g. the actual location of the subscriber.

References

- [1] IN Switch Solutions, <http://www.inswitch.us>
- [2] TribalText Studio 2003, <http://www.tribalttext.com>
- [3] ETSI. Digital Cellular Telecommunications System (Phase 2+, Release 98). Subscriber Identity Module Application Programming Interface (SIM API). Service description, Stage 2. Sophia Antipolis, France, 1999.
- [4] ORGA Systems enabling services GmbH. wireless Information Query. <http://www.orga-systems.com> -> Messaging Services -> Platform
- [5] The JBoss community. <http://www.jboss.org>
- [6] T. Janssen. *Konzeption und Realisierung eines grafischen Creation Environment für interaktive, menügetriebene Mobilfunkwendungen*. Diplomarbeit, Fachhochschule der Wirtschaft, Paderborn, Germany, 2004 (In German)
- [7] J. Eberspächer, H.-J. Vögel, C. Bettstetter. GSM Global System for Mobile Communication, B.G. Teubner, Stuttgart, Leipzig, Wiesbaden, 2001