

A New Framework for Indicating Terminal Capabilities in the IP Multimedia Subsystem

Bassam El Saghir Noel Crespi
GET / Institut National des Telecommunications
9 rue Charles Fourier - 91011 EVRY - France
{bassam.el_saghir, noel.crespi} @int-evry.fr

Abstract- The standardization of the IMS as a multi-access network implies that it can be accessed by different classes of terminals. Even at the heart of the same class of terminals (mobile phones for example), we are witnessing an increasingly divergent range of hardware and software capabilities (screen size, memory, OS). As a result of this heterogeneity, users may receive content that is not compatible with their device capabilities, and may even be unable to display this content. In order to ensure a better service and an optimal user experience, terminals should be able to communicate their capabilities to the network. The network could then use these capabilities to adapt its services and it may communicate them to other users as well. In this paper, we establish a state of the art concerning existing methods for indicating terminal capabilities in IMS. This state of the art will be used to design the architecture and the necessary mechanisms to implement this service in IMS, while taking into account the particularity of the IMS architecture and the possibility of integrating this service with Presence.

I. INTRODUCTION

There is a widespread agreement that convergence in the telecommunications world is becoming more and more a reality. This convergence is occurring simultaneously at the service level, terminal level and network level, and is driven mainly by current technological and market trends [1] as well as expected user needs.

At the service level, people demand rich-content and bundled services that extend well beyond voice to include text, image, audio and video. These services include PoC (Push over Cellular), IM (Instant Messaging), MMS (Multimedia Messaging Service), video streaming, online gaming and access to various multimedia content such as news, sports and music. Moreover, people want their services to be delivered in a seamless and personalized way, e.g. anywhere, anytime, using any available device and whatever access network is appropriate. This paves the way for a new generation of services with mobility, device and access awareness.

At the terminal level, convergence is made possible thanks to recent advances in electronics and miniaturization. Terminals which were historically optimized for one or few specific tasks (e.g. traditional cell phones) are becoming increasingly multifunctional as they encompass multiple devices: mobile phone, PDA, game console, music player, video camera, radio, GPS and TV receiver. This in turn adds to the complexity of their hardware and software capabilities.

At the network level, convergence implies the delivery of services to end users with a suitable quality of service regardless of the access types used. Nowadays, these access types are more diverse than ever with wireless and wire line technologies such as GSM, GPRS, UMTS, WLAN, xDSL and cable. Naturally, these technologies complement each other as each one is suitable for a different situation (e.g. xDSL at home/office, WLAN at a specific hotspot, GPRS/UMTS on the move). From an end user perspective, achieving true network convergence implies that he can benefit from access independent services such as being able to access emails via WLAN or GPRS or playing a game over Bluetooth with a friend in the same room and then inviting a far-away friend to join the game via UMTS.

The need for a converged solution has led the 3GPP to standardize the IP Multimedia Subsystem (IMS), a SIP based multi-layered architecture for providing mobile multimedia services. The IMS architecture was designed to ensure network convergence, since it consists of a unified core network with access networks that complement each other and common service delivery platforms. Moreover, it allows service convergence since the service platforms are access-independent and flexible enough to allow the easy integration of new services, while eventually reusing existing service components (service enablers).

However, IMS does not fully address the terminal convergence issue. Although its status as a multi-access network allows it to be accessed by a wide variety of terminals (mobile phones, PCs, PDAs, smartphones), it does not provide any mechanisms for retrieving hardware and software information about them. Such information gathering would be necessary if services are to be adapted to the heterogeneous capabilities of today's and future terminals. There are many applications that could benefit from terminal capabilities indication:

- *Intelligent routing*: a session can be automatically routed to the terminal which handles it the best. For example, a video session is preferably routed to a videophone.
- *Feature preview*: the availability (or absence) of features in the callee's terminal could be made available to the caller before session establishment. For example, all video related call options are deactivated (grayed out) on the caller screen if the callee's terminal does not support video.

- *User preferences awareness*: There are many user preferences that affect the perceived capabilities of his terminal, for example deactivating sound to save battery power. Such preferences can be taken into account by translating them as modifications in terminal capabilities.
- *Software installation and update*: Hardware and software capabilities information could be used to determine if an application runs correctly on a particular terminal. If not, the downloading of the application could be conditioned by the installation of the appropriate updates or patches that are not listed in the capabilities information.
- *Hardware changes*: Terminal capabilities information can reflect changes made by the user to the default hardware characteristics of his terminal, whether they are permanent (upgrading memory) or temporary (connecting a terminal to a large screen).

In this paper, we establish a state of the art concerning the existing methods for describing terminal capabilities. Then, we propose a complete solution for indicating terminal capabilities in IMS, while taking into account the particularity of the IMS architecture and the possibility of integrating this service with other services such as Presence. In the last section, we extend our solution to allow commonly used SIP messages to carry terminal capabilities information.

II. EXISTING METHODS

Many solutions have been proposed for indicating terminals capabilities [14]. The only solution already compatible with IMS is described in RFC 3840 [8]. This RFC describes the mechanisms by which a Session Initiation Protocol (SIP) user agent [12] communicates its capabilities and characteristics to other user agents and to the registrar for its domain. This information is conveyed as parameters of the Contact header field in many SIP messages (REGISTER, INVITE, OK reply to OPTIONS). Among the parameters directly related to terminals capabilities are audio, video, data, class, automata, duplex, and mobility. Although it was cited in 3GPP technical specifications as a possible solution for indicating terminal capabilities in IMS, this RFC does not offer a complete solution because its vocabulary is too restrictive and the profiles cannot be flexibly manipulated as in the solutions cited afterwards.

Another solution was proposed through the Composite Capability / Preferences Profile (CC/PP), a standard created by the World Wide Web Consortium (W3C) [9]. CC/PP is a comprehensive method for communicating the capabilities of devices such as clients, proxies, gateways and caches as well as resources such as documents. It is based on the Resource Description Framework (RDF), another standard created by the W3C [15], and XML. CC/PP presents many advantages. First, it is not restricted to specific applications; rather it is a generic language for constructing vocabularies for device capabilities, meaning that it can be used for various applications ranging from display and printing devices to proxy servers. Second, it supports the composition of preferences and profiles originating from multiple sources

(e.g. hardware vendors, software vendors, users, etc.). Third, it allows the use of indirect references (URLs) to refer to collections of capabilities (or attributes), thus providing flexibility in profile fetching and caching and allowing the composition of lightweight profiles (useful in low bandwidth networks). Still, CC/PP does not provide a standard vocabulary for describing device capabilities.

The aforementioned drawback was addressed by the WAP User Agent Profile (UAPROF), a standard started by the WAP Forum and actually developed by the Open Mobile Alliance (OMA) [10]. It extends the WAP 1.1 specification by adding a new profile known as Capability and Preference Information (CPI). UAPROF is an application of CC/PP, with the only difference that it provides a standard vocabulary for WAP clients to communicate their capabilities to servers. Therefore, UAPROF has all the advantages that were previously mentioned for CC/PP, plus the advantage of an already comprehensive and extensible vocabulary. UAPROF defines five different categories of device capability: software, hardware, browser, network and WAP. Therefore, it provides information about the capabilities of the network as well as the capabilities of the device.

Other standards have been proposed for different purposes such as synchronization and interconnection, but they incorporate nevertheless methods for describing device characteristics. One of these standards is SyncML which aims at developing a common synchronization protocol for data between mobile devices such as phones and PDAs. During synchronization, these devices exchange a description of their capabilities using the SyncML Device Information standard (DevInf) which is implemented directly in XML [13]. Another standard is Universal Plug and Play (UPnP) developed by Microsoft for device independent interconnection. It uses XML in order to provide a general method for describing device capabilities, but it does not specify any vocabulary as it is expected that this will be done by device manufacturers.

III. PROPOSED SOLUTION

In this section, we describe a solution for indicating terminal capabilities in IMS. We consider three aspects: capabilities description, network architecture and signaling mechanisms.

Concerning the capabilities description, we have adopted the UAPROF framework [10] which combines the advantages of CC/PP (flexibility and extensibility, profile composition and manipulation, use of indirect references) with a fairly large and generic vocabulary (although it was originally designed for WAP phones). Therefore, terminal capabilities information is structured into UAPROF compatible profiles which are integrated in the bodies of SIP messages. The construction and processing of such profiles will be described later in this paper.

As for the architecture and signaling mechanisms, they were based on the 3GPP specifications of the Presence service, mainly TS 23.141 [4] and TS 24.141 [2]. The reasons behind this approach are the following:

- Both services have a common goal of indicating the state and the availability of an object, whether it is a user (for Presence) or terminal capabilities (for terminal capabilities signaling). Therefore, the solutions proposed for both services must be similar.
- Terminal capabilities could be considered as a subset of user state information and implemented as part of the user presence profile. Therefore, the perspective of integrating terminal capabilities indication in Presence is highly conceivable and implies an acceptable degree of compatibility between their architectures and signaling mechanisms.

IV. ARCHITECTURE AND FUNCTIONAL ENTITIES

The architecture for terminal capabilities signaling is illustrated in Fig. 1. Each of the entities and reference points in this architecture has its counterpart in the Presence architecture as presented in the TS 23.141 specification [4], making an eventual combination of the two architectures rather intuitive. The watcher proxy and the terminal capabilities proxy have the same functions as the watcher proxy and the presence proxy in the Presence specifications. The other entities, however, present some functional differences, as it appears from the entities descriptions below.

A. Terminal Capabilities User Agent (TCUA)

This agent is located in the terminal itself and is responsible for gathering its capabilities and transmitting them to the network. Its main functions are described below.

A.1 Communication of capabilities information

When the TCUA intends to communicate its capabilities information, it generates a SIP PUBLISH request [5] targeted at the Terminal Capabilities Server (TCS) which is responsible for storing and managing this information. The PUBLISH request is generated in accordance with RFC 3903 [5] and should contain an “Event” header with a value reflecting the Event Package associated with terminal capabilities. The request body contains a capabilities profile which consists of a XML document structured according to the UAPROF specification. The vocabulary used by this profile could be the same as that of UAPROF. The TCUA must implement the “application/rdf+xml” MIME type as described in RFC 3870 [6]. The PUBLISH request could be generated either periodically, on request from the TCS, or when changes occur in terminal capabilities.

A.2 Communication of partial capabilities information

The TCUA does not always need to transmit all its capabilities information each time it needs to update this information to the TCS. For example, if changes in terminal capabilities are small, transmitting the whole profile creates an unnecessary overhead which could be critical in networks with restricted bandwidth, as in today’s wireless networks. Therefore, the TCUA should be able to build partial profiles by including only relevant information (i.e. changes from the last transmitted profile).

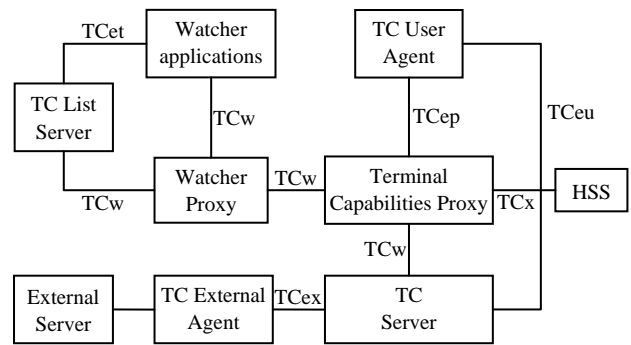


Figure 1. Reference Architecture

There are two other cases where the use of partial profiles is beneficial: when the TCUA can not get all capabilities information from the terminal or when part of the information could be transmitted by a third entity whose connection to the TCS is better than the TCUA connection (a frequent case when the TCUA resides in a wireless terminal). In these cases, complementary information is provided by a new entity who communicates with the TCS through a Terminal Capabilities External Agent (TCEA).

Presence specifications have already adopted drafts specifying extensions and mechanisms for creating partial profiles. However, these drafts introduce much complexity which is not necessary for information such as terminal capabilities. As a result, we adopted a much simpler approach similar to the one defined in the UAPROF specification, but with slight modifications. This approach is described in the TCS section.

B. Terminal Capabilities External Agent (TCEA)

This agent is responsible for collecting capabilities information from external servers and transmitting it to the TCS. This information could be default profiles provided by terminal manufacturers and application developers through their own servers. Default profiles could be defined for hardware (terminals and peripherals) or software (operating systems and applications).

The TCEA acts as a translator between external servers and the TCS. It communicates with external servers using their own protocols (HTTP, WSP ...) and transmits their information to the TCS. Therefore, it should also implement the TCUA functions as described in the previous section. If the external server use UAPROF for coding default profiles, the role of the TCEA consists simply of extracting the UAPROF profile and copying it into a PUBLISH message destined to the TCS.

C. Terminal Capabilities Server (TCS)

The TCS is a SIP application server whose main functions are storage, management and distribution of terminal capabilities information of users belonging to his domain. These functions are described below.

C.1 Subscription acceptance from watchers

The TCS must be able to handle subscriptions from watchers to capabilities information of TCUsAs. (See section D.) When a TCS receives a SUBSCRIBE request from a watcher concerning particular information of a TCUA, it attempts to verify the watcher identity and perform authorization in the same way as the Presence Server in 3GPP TS 24.141 [2] subclause 5.3.3.2. If the subscription is successful, the TCS shall generate a response to the SUBSCRIBE request and notifications in accordance with the same subclause.

C.2 Publication acceptance of terminal capabilities information

Like the Presence Server, the TCS acts as an Event State Compositor (ESC) according to 3GPP TS 24.141 [2] subclause 5.3.3.3. When the TCS receives a PUBLISH request from the TCUA (or a TCEA), it attempts to verify the watcher identity, perform authorization and process the PUBLISH request according to the same subclause (Fig. 2).

C.3 Management of terminal capabilities profiles

The TCS receives capabilities information about a particular TCUA through PUBLISH messages. These messages originate from the TCUA itself or from one or multiple TCEAs which have additional information about this TCUA. The first PUBLISH message sent by the TCUA to the TCS should contain enough information to allow the latter to build an “initial profile” of the TCUA. The initial profile is defined as a capabilities profile containing all possible information about the TCUA’s terminal capabilities and used by the TCS to apply subsequent updates and changes. If the TCUA already provides a complete profile in its first PUBLISH message, the TCS stores it as the initial profile without further processing.

However, if the TCUA provides a partial profile, the TCS builds the initial profile of the TCUA as follows (Fig. 3):

- The TCS extracts all the reference URLs from the partial profile and contact the TCEAs to retrieve the default

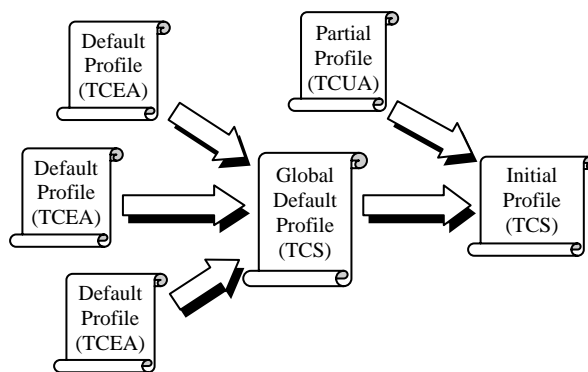


Figure 3. Building of the initial profile by the TCS

profiles referenced in these URLs. If there are no URLs in the profile, the TCS tries to infer them from information about terminal, such as the IMEI identifier for mobile phones which gives information about the manufacturer and the model of the phone, and therefore determines the external server and the default profile that should be extracted from it.

- The TCS combines all the default profiles it has retrieved from TCEAs into one single profile known as “global default profile.”
- The TCS uses the partial profile provided by the TCUA to overwrite the global default profile. This means that any attribute in the global default profile which also exists in the partial profile has its value replaced with the value in the partial profile, and any attributes that exist only in the partial profile are added to the global default profile. The modified global profile becomes the initial profile which is stored by the TCS.

When the TCS receives a subsequent PUBLISH request, it extracts the capabilities profile from the request, compares it to the stored profile and applies the corresponding changes to it. Each of these changes is applied according to the resolution rule associated with each attribute. Three resolution rules are defined in the UAPROF specification [10]:

- Locked: The final value of the attribute is determined by its first occurrence outside the default description block (subsequent occurrences are ignored). In other words, the value of the attribute in the initial profile is always retained in the updated profile. This resolution rule is useful for attributes that cannot be modified by users, but by manufacturers and developers. For example: phone model, processor speed.
- Override: The final value of the attribute is determined by the last occurrence of that attribute. In other words, this attribute is always updated with the last value sent by the TCUA. This is typical for attributes controlled by the user. For example: screen resolution, number of colors.
- Append: The final value of the attribute is a list of all the values of that attribute. This means that the new value of the attribute is appended to the list of existing values. For example: Supported codecs, Supported MIME types.

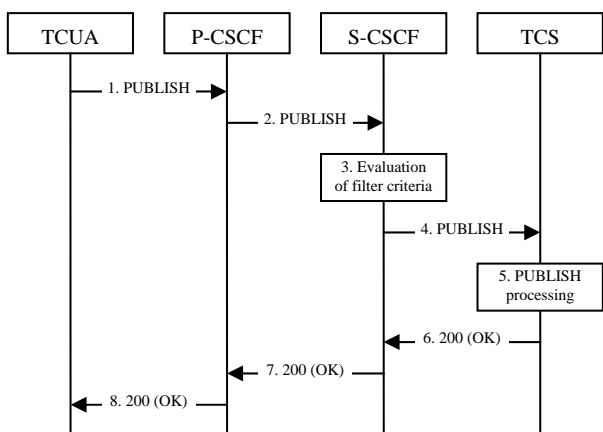


Figure 2. Signaling flow for TCUA publication

C.4 Notification of terminal capabilities changes

The TCS generates notifications of changes to the watchers using SIP NOTIFY messages in accordance with RFC 3265 [11]. The first NOTIFY message to a watcher should be generated immediately after its subscription and should contain all the TCUA capabilities information that the watcher has subscribed to (Fig.4). This allows the watcher to build his own initial profile of the TCUA, and therefore, to support partial notifications from the TCS. The notifications from the TCS to the watcher could be generated either periodically, on request from the watcher, or when changes occur in the terminal capabilities of the concerned TCUA.

D. Watcher

A watcher is an entity whose main purpose is gathering capabilities information about one or multiple TCUs. He could be located in a user terminal or in an application server. Its main functions are the following:

D.1 Subscription to terminal capabilities information and processing of notifications

The watcher subscribes to terminal capabilities information of a particular TCUA by sending a SIP SUBSCRIBE request to the TCS in accordance with RFC 3265 [11] and RFC 3856 [7]. Once the subscription is successful, it can receive complete or partial profiles through NOTIFY messages sent by the TCS. When the watcher receives a NOTIFY message for information about a particular TCUA, it extracts the profile from the message body and overwrites the current profile that it has stored for this TCUA. This means that any attribute in the current stored profile which also exists in the received profile has its value replaced with the value in the received profile, and any attributes that exist only in the

received profile are added to the current profile.

D.2 Subscription to capabilities information of a list of TCUs

The watcher can subscribe to capabilities information of a list of TCUs by sending a single SUBSCRIBE request to a Terminal Capabilities List Server (TCLS). The target of the subscription is a SIP URI representing the list of TCUs that the watcher needs information about. The watcher should be able to create and manage his own lists of TCUs through the TCLS.

E. Terminal Capabilities List Server (TCLS)

The TCLS is a SIP application server whose main functions are storing watchers' lists of TCUs and managing their subscriptions to these lists. These functions are described below:

E.1 Subscription acceptance from watchers

The TCLS must be able to handle subscriptions of watchers in the same way as the TCS. When the TCLS receives a SUBSCRIBE request from a watcher for information about a TCUA list (represented by a SIP URI), the TCLS attempts first to verify the watcher identity and perform authorization in the same way as the Presence Server in 3GPP TS 24.141 [2] subclause 5.3.3.2. Then, it finds the list corresponding to the SIP URI and, for each TCUA in this list, generates subscription requests to the TCS on behalf of the watcher. When all subscriptions are complete, the TCLS generates a response to the SUBSCRIBE request of the watcher according to the same subclause. It also relays to the watcher the first notification messages from the TCS containing the full profiles of the TCUs.

E.2 Notification of terminal capabilities changes

When the TCLS receives notification messages from the TCS, it can relay them one by one to the watcher or it can extract the profiles and aggregate them in one NOTIFY message which will be sent to the watcher at a specific time. In order to aggregate multiple profiles in a single message, the TCLS and the watcher must implement the "multipart/related" MIME type as described in RFC 2387 [16].

V. EXTENSION TO OTHER SIP MESSAGES

The solution that we proposed in the previous section suggests that the TCUA communicates its capabilities profile to the TCS using only one message, the PUBLISH message. However, it is possible to include the profile in the body of other SIP messages that are being transmitted to the network, thus avoiding the transmission of separate PUBLISH messages.

In order to implement this alternative, the SIP messages must be redirected to the TCS before they reach their destinations. This is done by defining an additional iFC (initial Filter Criteria) in the S-CSCF which is located in the same domain as the TCS. This iFC is stored in the Home

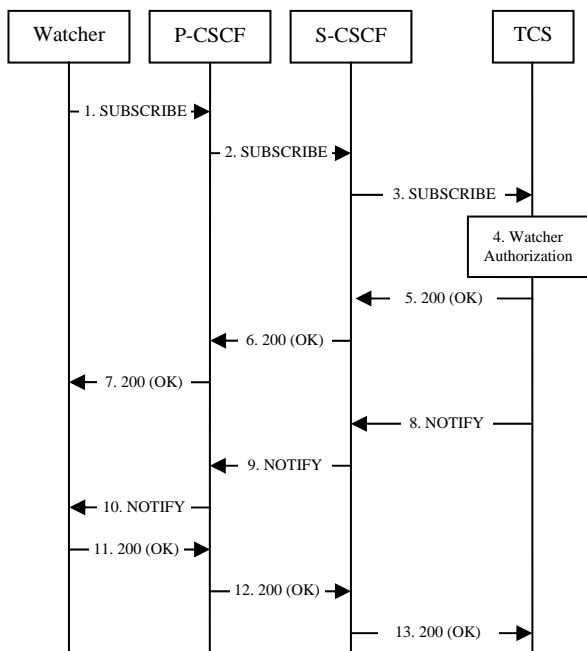


Figure 4. Signaling flow for watcher subscription and TCS notification

Subscriber Server (HSS) along with the other elements of the profile of the user represented by the TCUA.

The iFC definition is shown in Table 1. According to this definition, all SIP messages originating from the TCUA and containing in their bodies a document with an “application/rdf+xml” MIME type are redirected to TCS. We make the following notes:

- This iFC relies on the presence of the “application/rdf+xml” MIME type in order to identify a SIP message with an embedded capabilities profile. This works if terminal capabilities indication was the only application using RDF formatted documents in IMS. If this is not the case, a new MIME type must be defined to identify RDF documents describing terminal capabilities, for example: “application/tc+rdf+xml.”
- According to our solution, SIP messages used to carry capabilities profiles must be able to trigger the evaluation of iFCs when they reach the S-CSCF. Therefore, SIP messages that do not trigger iFC evaluation (such as NOTIFY, PRACK, UPDATE and BYE), could not be used.
- The TCUA should implement the “multipart/mixed” MIME type as defined in RFC 2046 [17] if it wants to carry its capabilities profile in a SIP message which has already its own message body (such as SDP information in INVITE).

When the S-CSCF receives a SIP message, it evaluates all the iFCs including the terminal capabilities iFC. If the SIP message contains a capabilities profile, the S-CSCF redirects it to the TCS which extracts the capabilities profile from the message. Since the capabilities profile will not be of use to other entities, the TCS removes the profile from the message before returning it to the S-CSCF. By applying these modifications on the SIP message, the TCS acts as a SIP B2BUA in accordance with 3GPP TS 23.218 [3] subclause 9.1.1.4, with the difference that it does not modify SIP header contents (except the “Content-type” header).

TABLE I
INITIAL FILTER CRITERIA SETTINGS

Field	Value
Session Case	Originating
SIP Method	*
Public User Identity	[TCUA public user identity]
Header : Content	Content-type : application/rdf+xml
Application server	[TCS public user identity]

VI. CONCLUSION AND FUTURE WORK

In this paper, we proposed a solution for indicating terminal capabilities in IMS. We first reviewed previous

methods for describing terminal capabilities and we chose the UAPROF technology as the most convenient solution for building capabilities profiles. Then, we used the current 3GPP specifications of Presence to propose architecture and mechanisms that are compatible with this service. We also showed how to transmit capabilities profiles using SIP messages other than PUBLISH. As future work, we plan to integrate our solution into Presence in order to create a “Rich Presence” information service in IMS. We will also extend our solution to communicate context information, which we consider as a basic step towards implementing context awareness and ambient intelligence in IMS.

REFERENCES

- [1] 3G Americas, “Convergence: An Outlook on Device, Service, Network and Technology Trends,” July 2005.
- [2] 3GPP TS 24.141, “Presence service using the IP Multimedia (IM) Core Network (CN) subsystem; Architecture and functional description; Stage 3 (Release 6),” v6.4.0, June 2005.
- [3] 3GPP TS 23.218, “IP Multimedia (IM) session handling; IM call model; Stage 2 (Release 7),” v7.0.0, June 2005.
- [4] 3GPP TS 23.141, “Presence service; Architecture and functional description; Stage 2 (Release 6),” v6.8.0, June 2005.
- [5] A. Niemi, “Session Initiation Protocol (SIP) Extension for Event State Publication,” RFC 3903, October 2004.
- [6] A. Swartz, “Application/rdf+xml Media Type Registration,” RFC 3870, September 2004.
- [7] J. Rosenberg, “A Presence Event Package for the Session Initiation Protocol (SIP),” RFC 3856, August 2004.
- [8] J. Rosenberg, H. Schulzrinne, and P. Kyzivat, “Indicating User Agent Capabilities in the Session Initiation Protocol (SIP),” RFC 3840, August 2004.
- [9] REC-CCPP-struct-vocab, “Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0,” W3C Recommendation, 15 January 2004, <http://www.w3.org/TR/CCPP-struct-vocab/>.
- [10] OMA-UAPROF, “User Agent Profiling Specification (UAPROF) 1.1,” Open Mobile Alliance, 12 December 2002, http://www.openmobilealliance.org/release_program/docs/UAProf/OMA-WAP-UAProf-V1_1-20021212-C.pdf.
- [11] A. B. Roach, “Session Initiation Protocol (SIP) Specific Event Notification,” RFC 3265, June 2002.
- [12] J. Rosenberg et al., “SIP: Session Initiation Protocol,” RFC 3261, June 2002.
- [13] SyncML-DevInf, “SyncML Device Information DTD version 1.1,” Open Mobile Alliance, 15 February 2002, http://www.openmobilealliance.org/tech/affiliates/syncml/syncml_devinf_v101_20010615.pdf.
- [14] M. H. Butler, “Current Technologies for Device Independence,” Publishing Systems and Solutions, Laboratory, HP Laboratories Bristol, 4 July 2001, <http://www.hpl.hp.com/techreports/2001/HPL-2001-83.pdf>.
- [15] CR-rdf-schema, “Resource Description Framework (RDF) Schema Specification 1.0,” W3C Candidate Recommendation, 27 March 2000, <http://www.w3.org/TR/rdf-schema>.
- [16] E. Levinson, “The MIME Multipart/Related Content-type,” RFC 2387, August 1998.
- [17] N. Freed and N. Borenstein, “Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types,” RFC 2046, November 1996.