Ambient Community Services – Concept and Implementation

Konstantinos DEMESTICHAS[1], Evgenia ADAMOPOULOU[1], John MARKOULIDAKIS[2], and Michael THEOLOGOU[1]

Abstract—**The present paper introduces the innovative concept of a new type of services, called Ambient Community Services (ACS). The ACS concept utilizes information that resides in communities of mobile terminals, such as location, speed, sensor-related data or user-created multimedia content. After the collection, processing and filtering of these data, a number of innovative, heterogeneous value-added applications can be provided to the end-users, such as the provision of vehicular traffic conditions. ACS advances the social networking by allowing users to form cooperative, service-enabling communities. This paper includes a thorough description of the state-of-the-art, the proposed system architecture and communication flow, potential enabled applications, important implementation aspects, as well as future research activities.**

Index Terms—**Community services, Mobile services, Mobile terminals, Real-time applications**

1 INTRODUCTION

THE current mobile communications landscape is characterized by tremendous progress in the radio access technologies supported (2G/2.5G/3G, WLAN, DVB-T, etc.). This network-related evolvement, however, is tightly accompanied by a significant advancement in the field of mobile terminals. The increased capabilities of mobile devices, in terms of processing power, storage capacity, multimedia support, capturing capabilities etc., have set the path for expanding functionality from the network to the mobile terminal. In this sense, the mobile terminal can enhance its up to now role of being a service-delivery instrument to being a fully functional, service-enabling and content creation tool.

This remarkable increase in the capabilities of the mobile terminal, along with its pervasion in the market, has elevated the terminal's role into an irreplaceable, omnipresent device of the daily life. Consequently, a natural evolution is the use of the mobile terminal not only as a traditional communication-enabling device but also as an integral part of all expressions of the user's personal life. Towards this vision, the mobile terminal can undertake the role of enhancing the social networking. This paper introduces a novel type of services targeted for mobile users, namely the Ambient Community Services (ACS). The aim is to explore the transition from explicit communication into delivering and receiving information through context disclosure, by exploiting the underlying spatial and temporal patterns within a local community.

In particular, the ACS concept exploits the ability of the network to retrieve in real time useful data from multiple mobile terminals. The involved mobile terminals belong to specific communities. The collected data are impersonalized and can be generated either at the time of the request, or at an earlier time (in which case they remain stored in the terminal). Subsequently, the gathered data are processed (at the network side), in order to compose the target information or content that will be delivered to the requesting terminal.

Our vision is to build an open framework based on the ACS concept that will enable the provision of various types of innovative services to the end-users. Example services include: Real Time Vehicular Traffic Conditions, Clubbing/Restaurant Information, File Sharing, Environmental Applications, Mobile Multiplayer Gaming, Emergency Services, Socializing Applications, and others.

The remainder of this paper is structured as follows: Section 2 is concerned with the related state-of-the-art landscape. Section 3 presents in detail the proposed architecture for the deployment of ACS. Section 4 provides a list of interesting potential applications enabled by the ACS concept. Section 5 includes detailed implementation aspects and outlines the testbed that has been developed so far for ACS provisioning. Section 6 draws concluding remarks and summarizes important related future research activities.

2 STATE OF THE ART

The increased capabilities of modern mobile terminals, in terms of processing power, storage capacity, multimedia

Manuscript received Oct. 20, 2007, accepted Jan. 21, 2008. The work presented herein is conducted in the framework of Ph.D. research performed by K. Demestichas and E. Adamopoulou, under the supervision of Prof. M. Theologou. The work is also supported in part by the European Commission, under the Sixth Framework Program, in the context of the IST project MOTIVE (FP6-IST-27659).

[1]K. Demestichas, E. Adamopoulou and M. Theologou are with the National Technical University of Athens, School of Electrical and Computer Engineering, 9 Iroon Polytechniou Str., Zographou GR-157 73, Athens, Greece (tel.: +30 210 772 1493, e-mails: cdemest@cn.ntua.gr, eadam@cn.ntua.gr, theolog@cs.ntua.gr).

[2]J. Markoulidakis is with Vodafone-Panafon Greece, R&D Department, 1-3 Tzavella Str., Halandri GR-152 31, Greece, (tel.: +30 210 6702776, e-mail: Yannis.Markoulidakis@vodafone.com).

support, capturing capabilities, etc., have motivated the development of novel, feature-rich services, beyond the traditional telephony and SMS applications. In today's market, a considerable number of mobile value-added services have been introduced, including the purchase of ring-tones, wallpapers, games and videos, chatting, prize competitions, informative services, such as live scores, and so forth. A relatively new area of value-added services is that of location-based services [1],[7], which include user and fleet tracking, navigation, location-based advertising and billing. This relatively recent, advanced type of services reveals that context-related information, such as user location, will play an integral part in the delivery of value-added services to future mobile phones.

Another, even more recent, class of applications that attracts particular attention nowadays is the area of social mobile applications [10], the main notion of which is to expand the functionality and social interactions of services such as forums, chat-rooms, blogs etc., from the Internet to the mobile world. In this light, the mobile terminal –a device already omnipresent in everyday life– is beginning to undertake a new, central role within the context of social environment and non-work life. Towards this direction, a number of social mobile applications have been introduced in today's market:

- *"Rendezvous" coordination applications*: The main objective of this type of applications, e.g. Dodgeball [2], Playtxt [9], and Reno [10],[11], is to enable users to reveal their location to other users, so as to coordinate their leisure-time activities, meet and socialize. Reno, being developed by Intel Research Seattle [5], is thus far the most advanced service of this kind. With Reno mobile phone application users can query other users about their location, as well as disclose their own, either in response to another query or unprompted. Unlike other mobile social applications that support "rendezvous", Reno is location-aware. For example, Dodgeball and Playtxt require from the user to manually type in an identifier –a place name or postal code, respectively– and send these data to a central server that performs the location calculation. In contrast, Reno calculates the device's approximate location locally, using GSM positioning estimation techniques, and then presents the user with a short list of nearby locations sorted by proximity.
- *"Moblogs"*: A mobile weblog, or "moblog", consists of content posted to the Internet from a mobile or portable device, such as a mobile phone or PDA. Moblogs, such as Flickr [3], TextAmerica [13], MoblogUK [6], and Zannel [16], enable users to share multimedia files, especially photos captured via mobile terminals, with their friends, family and other community members on-line, where the pictures may be collaboratively annotated. In the same spirit, Yospace [15] provides tools that enable users to set up on-line collections of multimedia content. Once a user has set up their collection, they can invite guests to view their albums on-line and vote or comment on them.
- *Geo-referenced photo sharing*: By augmenting image capture with location-awareness, a new type of moblogs and community Web sites has been developed. This type of community-based sites, such as Socialight [12], goes one step further, by letting users share "geo-tagged sticky notes" created via their mobile phones. These notes are conveniently displayed on maps, and users may browse through them even without subscribing. A note usually includes the originator's opinion, recommendation or remark about the place he has visited (e.g., restaurant, bar, etc.), as well as an optional photo. Built-in geo-tagging support has also been added to Flickr, allowing users to view personal photo collections and public photo pools on a map. In combination with ZoneTag [17], a mobile phone application that can automatically tag photos with their location (based on the cell tower near which they were taken), as well as automatically upload them on Flickr, photo sharing can easily be done at the time of photo capture. Apart from this, however, research has also concentrated on investigating ways in which geo-tagging can further evolve. Ymogen [14] is an initiative of BBC Innovation regarding citizen journalism, aiming at combining GPS data with mobile phone capabilities. Activities include the creation of online galleries intended to "explore how people in the future might use mobile devices to capture and share multi-media stories". Ymogen intends to achieve more than just adding GPS data to mobile content. As an example, pictures and video captured from multiple devices around a sporting event can be mapped in time and space automatically, creating a rich, enhanced view of the entire game from multiple perspectives.

The ACS concept comes as a valuable extension to the state-of-the-art in the area of mobile community-based services. As opposed to existing approaches, the ACS concept introduces the following innovative features: *(i)* Ability of automatic (as opposed to manual) data collection and uploading; *(ii)* Instantaneous formulation of the group of terminals that should be queried; *(iii)* Real-time processing of the collected data; *(iv)* Anonymity and privacy of the service enablers (community of queried terminals); and *(v)* Provision of various, possibly diverse, services under a common umbrella.

3 SERVICE-ENABLING ARCHITECTURE

The delivery of ACS services requires the presence of a software client at the terminal side (community members), and the deployment of an ACS Enabling Server, as well as one or more Application Servers, at the network side. The preservation of user anonymity and the depersonalization of collected data can be conducted either by a separate Proxy Server or by a module integrated into the ACS Enabling Server encompassing the appropriate functionality. This second approach is adopted in the present paper.

Network-Side

The ACS Enabling Server is responsible for providing an open interface towards the Application Servers, receive and decode their requests, in order to issue the corresponding queries towards the members of the appropriate community. The queried terminals' responses are then aggregated, depersonalized and forwarded to the requesting Application Server. The issuing of the queries and the gathering of the responses is carried out in real-time, i.e. at the time of the Application Server's request.

Correspondingly, the ACS Enabling Server comprises four functional modules, as depicted in the architecture of Figure 1: *(i)* The AS-Com Module, which receives requests from and transmits responses back to the Application Servers. *(ii)* The MC-Com Module, which is responsible for issuing queries to mobile community (MC) members and for collecting their responses. *(iii)* The Control Module, which analyzes the requests of the Application Servers, in order to select the appropriate query targets and form the appropriate query messages. The Control Module orchestrates the overall service-enabling process. It also performs the depersonalization of data once they have been collected from the queried users. For this reason, it is responsible for replacing the MSISDN (Mobile Station International ISDN Number) of each queried terminal with an alias, or even completely removing the MSISDN (two modes of operation). *(iv)* The Local Database Management Module, which is delegated to retrieve information from the database that contains the list of registered users and their profiles.

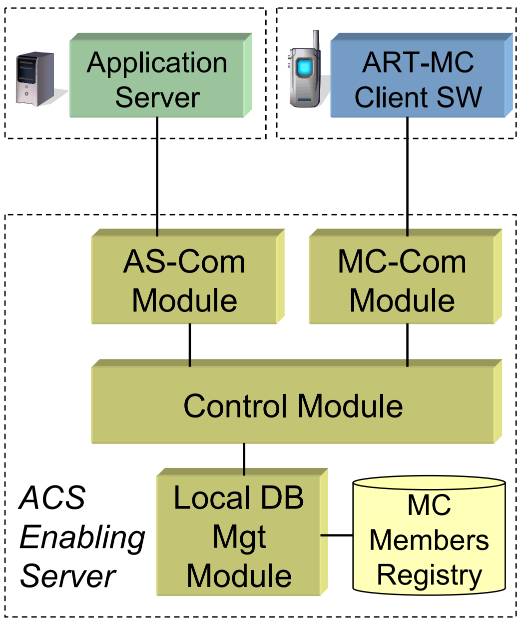

Figure 1: Architecture of the ACS Enabling Server

Terminal-Side

The ACS Terminal Client is responsible for awaiting query triggers from the ACS Enabling Server and responding back with the appropriate data. These responses may include either implicit or explicit user data. The first category may include data that do not require the user's intervention to be collected, such as the terminal position, velocity, perceived signal strength, information retrieved from integrated sensors (e.g., temperature, humidity, etc.), multimedia files (e.g., pictures, audio, video), etc. The second category includes data that need the user's intervention to be produced, such as ratings of restaurants, clubs and other places, personal opinions, suggestions and recommendations (related to the query messages), instructions (e.g., how to get to a place, what is the meaning or translation of a word or phrase, etc.), real-time multimedia content (e.g., "what are you seeing now?" through the terminal's video camera, etc.).

Correspondingly, the ACS Terminal Client comprises four functional modules, as depicted in the architecture of Figure 2. *(i)* The Activation Module, which waits for the ACS Enabling Server's querying messages and triggers the Control Module whenever such a message is received. *(ii)* The Control Module, which translates the incoming queries, so as to determine which local parameters are being requested. The Control Module orchestrates the overall process. *(iii)* The Local Parameter Retrieval Module, which retrieves the requested data (for example, location, speed, etc.). This module is also responsible for requesting from the user his/her intervention, whenever explicit user data are being requested. *(iv)* The Response Module, which encapsulates the retrieved data in a standard format and delivers them to the ACS Enabling Server.

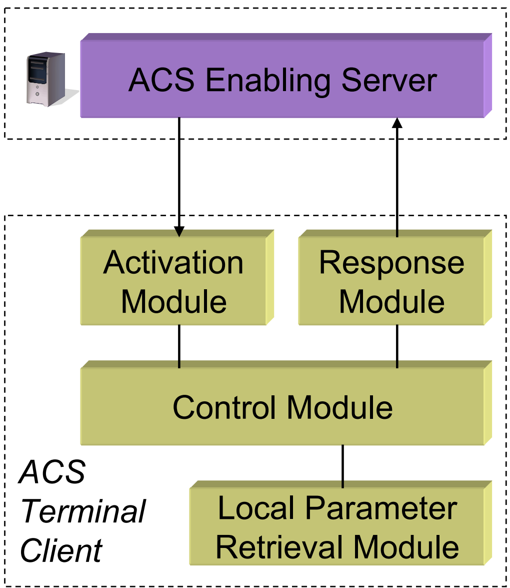

Figure 2: Architecture of the ACS Terminal Client

Communication Flow

Figure 3 presents a typical ACS communication flow. Four entities take part: the service requesting terminal (Service Requestor), an Application Server, the ACS Enabling Server, and the community of mobile terminals (Mobile Community). Initially, the requesting terminal opens a service layer session with an Application Server and requests a service.

Subsequently, the Application Server requests from the ACS Enabling Server to issue an appropriate query for impersonalized information. This communication follows the web services paradigm, over secure HTTPS connections. The ACS Enabling Server processes this request, in order to identify the target mobile community and form the appropriate query towards its members. Queries are issued either using a point-to-point SMS or WAP-push message, or using cell broadcast. The ACS Terminal Client of the queried mobile terminals handles the incoming query, by retrieving the requested local parameters and relaying them back to the ACS Enabling Server. The data upload process is carried out either through SMS or by opening a short data session (over GPRS, UMTS, or WiFi – whichever is available at the time). The ACS Enabling Server then depersonalizes the collected data, encapsulates them in a standard format and transmits them to the corresponding Application Server. The Application Server exploits the received data, in order to produce the information that was requested by the end-user. Finally, it delivers the information produced.

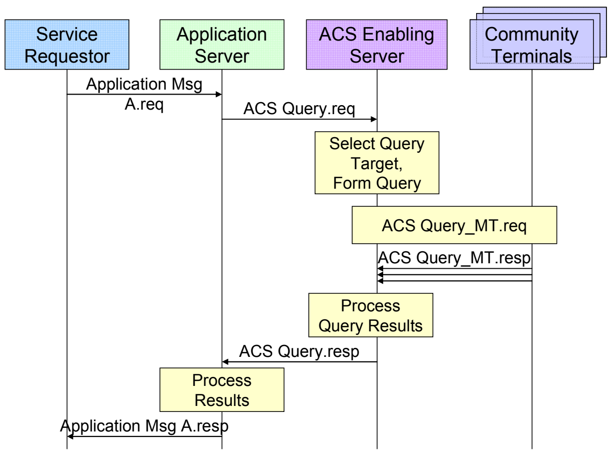

Figure 3: Typical ACS communication flow

4 ENABLED APPLICATIONS

The ACS concept can be exploited in various ways, enabling several heterogeneous applications. What follows is a list of interesting and representative examples:

- *Vehicular Traffic Conditions*: This service deals with the provision of real-time information concerning road-traffic conditions. The service requestor indicates the area/ road/ highway in question. The information that must be collected includes the position and velocity of the community members.
- *Clubbing/Restaurants Information*: This application is concerned with the provision of information regarding the popularity of clubs (or restaurants). Indicative questions that a service requestor may ask are: "What is the popularity of club C right now?", or "What is the popularity of club C over the past week?". Thus, in this use case scenario, the mobile community is a community of "clubbers". The information collected includes the positions of the community members, the time they spend in these positions, and optionally their explicit opinions or views about the clubs they visit.
- *File Sharing*: A file-sharing application may also be deployed using the ACS mechanism. This application will enable users to share content that is stored in their mobile terminals. The information that must be collected is the index of each user's shared files.
- *Environmental Applications*: Environmental applications may exist as well, e.g. informing end-users about the temperature or humidity levels in the areas of interest. The collected information includes the monitored environmental parameters, such as the temperature, that can be retrieved through integrated digital sensors.
- *Real-time Collection of Multimedia Content Related to Events*: ACS also concentrates on enabling the report of events or incidents by community members in real-time. Users who witness an important or serious event will be able to capture it through their mobile terminal and report it live to a network-side content server. Other members of the community will be able to view the content in real-time. Members, as well as non-members, will also be able to browse through the reported content, later on (not in real-time). This type of ACS can be conceived as a real-time mobile variant of the popular YouTube service. Some examples of events that may be reported by prospective "ACS youtubers" are the following:
  - sport events
  - music concerts
  - emergency incidents, such as road accidents, natural disasters, and so on ("citizen journalism")
  - places with monuments
  - places with beautiful scenery

In the event of serious or important incidents, an Application Server (e.g., a press agency) may request from the community members in the designated area to report (via their terminals' cameras and microphones) the situation. In this way, news is gathered faster, and can even assist the proper authorities in accurately assessing the situation. In addition, the exploitation of multiple perspectives of the same event can lead to the provision of enhanced, pluralistic content.

In general, a variety of heterogeneous applications may be supported by the proposed architecture; for example, mobile multiplayer gaming applications, chatting and socializing applications, emergency services, etc. may be provided as well. The ability to deliver diverse services and applications via the same framework is one of the main advantages of the proposed concept.

5 IMPLEMENTATION

Figure 4 depicts the testbed that has been developed for the implementation of ACS. The testbed comprises four entities: *(i)* the ACS Enabling Server; *(ii)* an Application Server; *(iii)* a community of mobile terminals; and *(iv)* a service requestor.

The two servers are hosted on separate computer systems running Windows XP Professional. They are connected to the Internet and have static public IP addresses. Each server has a GSM modem (in our case, a Sony Ericsson K700i phone connected via USB) attached, which simulates the functionality of an SMS gateway. In this way, the ACS Enabling Server is able to issue query messages through SMS, as well as gather the responses from the queried terminals (if sent by SMS). Additionally, the Application Server is able to receive requests from service requesting terminals, through SMS. The two servers are implemented using the Java 2 Enterprise Edition (J2EE) v1.4 platform. The implementation of the interface between the ACS Enabling Server and the Application Server is based on Web services over HTTPS.

The testbed's mobile community consists of four different phone models (Nokia N70, Nokia 6630, Nokia 6230 and Sony Ericsson P900). The implementation of the ACS Terminal Client relies on the Java 2 Micro Edition (J2ME) platform. The targeted devices are those supporting the J2ME platform, the Connected Limited Device Configuration (CLDC) v1.1, the Mobile Information Device Profile (MIDP) v2.0, as well as the optional Wireless Messaging API (JSR-120 or JSR-205) (JSR: Java Specification Request). Currently, these support requirements are covered by the majority of commercially available terminals. Further to this, it is expected that most future mobile phones will provide support for all of the aforementioned requirements, and that these features will be commonplace.

An advanced feature of MIDP 2.0 that has proved to be ideal for the implementation of ACS (at the terminal-side) is the Push Registry mechanism. "Push" is a very powerful concept and typically refers to the mechanism or ability to receive and act on information asynchronously, as information becomes available, instead of forcing the application to use synchronous polling techniques that increase resource use or latency [8]. The push registry enables J2ME applications to set themselves up to be launched automatically, without user initiation. The push registry manages network- and timer-initiated J2ME application activation; that is, it enables an inbound network connection or a timer-based alarm to wake a J2ME application up.

In combination with the Wireless Messaging API (WMA), it is possible to render a J2ME application capable of being automatically invoked whenever an inbound SMS message is received by the terminal. In order not to be invoked by all received SMS messages indiscriminately, the J2ME application is registered to be triggered only by SMS messages addressed to a specific port (or ports). In this way, normal SMS messages can be distinguished from messages that are to be handled by specific J2ME applications. The destination port number is encoded in the header of the SMS message.

Hence, for issuing an ACS query, the testbed's ACS Enabling Server sends out SMS messages targeted at a specific, pre-agreed port. On an ACS-enabled terminal, this SMS message is not directed to the terminal's native SMS inbox, but rather triggers the launch of the ACS Terminal Client (implemented in J2ME). The message is then decoded by the ACS Terminal Client, the requested local data are retrieved and relayed back to the ACS Enabling Server (either via SMS in case of short, textual data, or through a GPRS/3G/WLAN data session in case of large, binary data). It is important to notice that the ACS Terminal Client runs only when being invoked, for a short time, and then finishes; i.e., it does not run continuously, which results in a minimal consumption of resources.

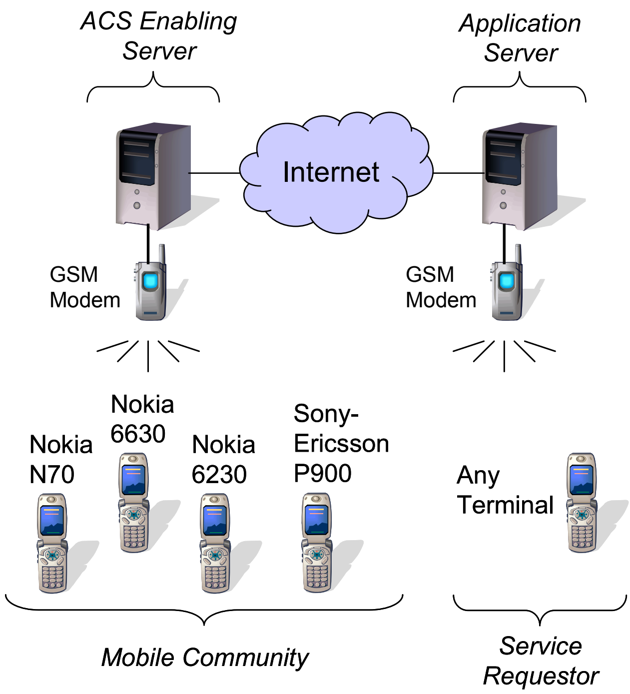

Figure 4: Configuration of the developed ACS testbed

Another significant implementation issue is that currently the J2ME platform does not provide an Application Programming Interface (API) for acquiring detailed information from a terminal device (e.g., the level of the perceived signal strength on the radio interface, or the battery level). This is expected to change in future specifications, for example in the upcoming MIDP 3.0 (JSR 271: Mobile Information Device Profile 3), which intends to provide support for extensive device capabilities query.

Nonetheless, the retrieval of crucial for the ACS concept local data remains still feasible thanks to the following J2ME APIs:

- JSR-179: Location API for J2ME (positioning information)
- JSR-253: Mobile Telephony API (e.g., Network ID, call-related phone and user parameters)
- JSR-82: Java APIs for Bluetooth (for communication and data retrieval from external Bluetooth sensors)
- JSR-75: PDA Optional Packages for the J2ME Platform (Personal Information Management (PIM) and FileConnection (FC) API, targeted for providing access to

file systems residing on mobile devices)

It is also possible for the ACS Terminal Client to pop-up a GUI so as to capture the user's explicit input, e.g. by presenting a text box or a list of choices.

In order to retrieve device information not available through the J2ME APIs, a developer has to resort to the native OS (e.g., Symbian, Microsoft Windows Mobile, etc.). An effective work-around solution is to build the Local Parameter Retrieval Module of Figure 2 as a native Server-Socket application [4] listening to a specific localhost port. The native application will provide support for the extraction of data not available through J2ME APIs, by allowing localhost socket clients to be connected and indicate what type of data they need to retrieve. The native application will then respond by retrieving the appropriate data and sending them as an array of bytes. In this communication process, the J2ME part of the ACS Terminal Client acts as the client-recipient of information, and the native part acts as the native data provider.

Regarding the testbed's service requestor, there are no specific requirements. A user may request an ACS service either by sending a specifically-formatted SMS message towards the Application Server or by visiting the Application Server's Web portal. Several different terminals have been tested, including Nokia's 6260, 6600 and E70 models, as well as Ericsson's K800. The current version of the testbed's Application Server offers support for the environmental application as well as for the vehicular traffic conditions application described in Section 4. The deployment of more Application Servers has been planned for the near future.

6 CONCLUSION

This paper introduced the innovative concept of Ambient Community Services (ACS). The state-of-the-art landscape was thoroughly presented and the differentiations of the ACS concept were analyzed. The proposed system architecture and the typical ACS communication flow were described. Several potential applications that can be enabled by the ACS concept were listed. Finally, detailed implementation aspects were discussed, and the implemented testbed's configuration was presented.

In general, the ACS concept utilizes information that resides in communities of mobile terminals, such as location, speed, sensor-related data or user-created multimedia content. After the collection, processing and filtering of these data, a number of innovative value-added applications can be provided to the end-users.

Further research activities will focus on:

- *The exploitation of short-range wireless technologies*. Specifically, we will aim at exploiting the integrated short-range connectivity capabilities of modern mobile terminals, in order to focus the ACS queries into specific, limited in size, sub-groups of users.
- *The combination of user-generated and commercial content*. The multimedia content that is uploaded by the ACS users may be distributed "as is", or alternatively it could be combined with the reports of other users from the same events, as well as with legacy commercial content, in order to produce an enriched outcome. For example, multiple reports from a sport event, together with the actual TV broadcasted content, may be correlated in order to generate feature-rich material containing views of the spectacle from multiple angles, 'behind the scenes' information, and numerous interviews.
- *The creation of hybrids between mobile and Internet communities*. ACS community members might, in the future, be not only mobile users, but plain PC users (connected to the Internet), as well. In this sense, the ACS concept intends to unite the two worlds, mobile and Internet, under a hybrid community scheme. PC users will be able to distribute their own generated content to mobile users and vice versa. In this way, the target group of ACS is further enlarged, encompassing both mobile and non-mobile users.

7 REFERENCES

[1] D'Roza, T., and Bilchev, G. An overview of location-based services, *BT Technology Journal, Kluwer Academic Publishers, 21*, 1 (Jan. 2003), 20-27.

[2] Dodgeball official Web site, http://www.dodgeball.com/.

[3] Flickr official Web site, http://www.flickr.com/.

[4] Gupta, A., and Jode, M. Extending the Reach of MIDlets: how MIDlets can access native services. Symbian Developer Network, Version 1.1 (June 2005). Article available on-line at http://developer.symbian.com/main/downloads/papers/MIDlet_Native_Services_Framework/MIDlet_Native_Services_Framework_v1.1.zip.

[5] Intel Research Seattle official Web site, http://www.intel-research.net/seattle/.

[6] MoblogUK official Web site, http://moblog.co.uk/.

[7] Mohapatra, D., and Suma, S. B. Survey of location based wireless services. In *Proc. of the IEEE International Conference on Personal Wireless Communications (ICPWC 2005)* (Jan. 2005), pp. 358-362.

[8] Ortiz, E. The MIDP 2.0 Push Registry. Article available on-line at http://developers.sun.com/techtopics/mobility/midp/articles/pushreg/ (Jan. 2003).

[9] Playtxt official Web site, http://www.playtxt.net/.

[10] Smith, I. Social-Mobile Applications, *IEEE Computer, 38*, 4, (April 2005), 84-85.

[11] Smith, I., Consolvo, S., Hightower, J., Iachello, G., LaMarca, A., Scott, J., Sohn, T., and Abowd, G. Social Disclosure of Place: From Location Technology to Communications Practices. In *Proc. of the International Conference on Pervasive Computing (Pervasive 2005)*. Springer Verlag, 134-151.

[12] Socialight official Web site, http://socialight.com/.

[13] TextAmerica official Web site, http://www.textamerica.com/.

[14] Ymogen official Web site, http://open.bbc.co.uk/labs/2006/london/ymogen/.

[15] Yospace official Web site, http://www.yospace.com/mcp_intro.html.

[16] Zannel official Web site, http://www.zannel.com/.

[17] ZoneTag official Web site, http://zonetag.research.yahoo.com/.