Design and Implementation of Mobile Electronic Memos: a tool to capture and share information in mobile environments

G. BARTOLOMEO[1], N. BLEFARI MELAZZI[1], S. SALSANO[1], R. WALKER[2]
[1] *Department of Electronic Engineering, Univ. of Roma "Tor Vergata", Roma, Italy*
Tel: +39 06 7259 7501, Fax: +39 06 7259 7435, Email: blefari@uniroma2.it
[2] *XiWrite s.r.l. Frascati, Italy*
Tel: +39 06 9417507, Fax: +39 06 9424742, Email: rwalker@xiwrite.com

Abstract: The first part of this paper introduces "MEMs" (Mobile Electronic Memos). MEMs are electronic notes consisting of a data structure associated with a specific class of information (e.g. information describing a location, a person, a service, or a Web site), and meta-information, used for management. Users can automatically capture MEMs from the environment or from other services, store them for future use, share them with other users and send them as input to other services and applications. With MEMs, users can drastically reduce the amount of information they have to input manually. MEMs should be seen as an extension of the traditional clipboard, a uniform mechanism allowing users to store output from one service and provide it to other services. We suggest that this possibility could play an important role in encouraging the take-up of mobile services by private and business users.

The second part of the paper describes the implementation of the MEM concept in the framework of a EU-funded project, named Simple Mobile Services. The implementation includes as main elements: i) an open source client for mobile devices (called MOVE), which simplifies access to mobile services and allows easy management of MEMs; ii) server-side components; iii) a middleware platform; iv) Smart Card Web Server (SCWS) technology to access the security features offered by the (U)-SIM, contained in the mobile devices. The whole solution has been implemented on real-life SIMs by a major operator, and is currently tested in real life environments on the University of Roma "Tor Vergata" campus and at Athens International Airport.

Finally, we point out that the aim of this paper is a high-level description of our work. More technical details can be found in project deliverables and documents. [1]

Keywords: Simple mobile services, context-aware services, location based services.

1. Information Capture and Storage for Mobile Users

Imagine a tourist travelling through London. Sometimes she needs information – for instance on how to reach a particular destination. Sometimes she needs to perform an action – like purchasing a theatre ticket. Well-designed mobile services serve these needs well. But although much of the necessary technology exists, current commercial services give little help in capturing information from the environment. If she sees the time of a show on a poster, all she can do is try and remember it, or write it down on a piece of paper, or perhaps on a PDA. If she wants to tell a friend to come and meet her at a certain café she

[1] This work was partially supported by the EU under the project IST2006-034620 "Simple Mobile Services" [1], and would not have been possible without important contributions from our colleagues in the S.M.S. project.

has to ask the owner for the address. And even when the information comes in electronic format it is usually unstructured (e.g. a web page with part of a train timetable, an email confirming an airline or a hotel booking).

This kind of information is hard to retrieve and share. If the user is not sending it immediately, she will have to search for it. And once she has found it, she will probably send it by email, forcing the recipient to find and interpret the relevant data. If the recipient wants to use the information in another program or give it to a third party service, she will have to "cut" it from its original location and "paste it" to where it is needed. In many cases she will need to parse the information (e.g. to see which parts of an address go into which field in an address book) or even translate it from one language into another. In fact, current mobile users receive information from a range of non-electronic and electronic sources: passers-by, posters, advertisements, presentations, phone conversations, email, the Web, text messages and a small number of dedicated mobile services. Where the information comes from a non-electronic source, the only way to store it on an electronic device is to input it manually into an application or document (e.g. an email address book, a note pad, a dedicated corporate application, a text document or spreadsheet). Where the source is electronic, the application that receives the information can store it in its own native format (though on the Web this can be problematic). However, there is no way of automatically copying something written in one format into a web page requiring it to be written in a different format. This kind of "intelligent" operation has to be performed by hand.

The manual and semi-manual techniques just described, are risk prone, clumsy, time-consuming and insecure. Memorized information can be forgotten; business cards and paper notes are easily mislaid; emails and stored web pages may be hard to find; copying is a common source of error. The authors of this paper believe that the ergonomic difficulties associated with manual input and copying of information are important obstacles to the uptake of mobile services.

The next generation of mobile services should try and overcome these problems. In particular they should make it easier for users to capture and reuse information from the environment and from other services. This is one of our goals in the European Simple Mobile Services (S.M.S.) project ([1], [2]). To reach this goal, the project introduced the concept of MEM (Mobile Electronic Memo). The goal of MEMs is to allow mobile users to capture and store information from the environment and from other services and to securely share this information with other applications, services and other users. Three scenarios that illustrate the potential of MEMs are described in [9].

1.1 Current solutions

To circumvent the difficulties described above, software manufacturers have proposed standard formats for specific kinds of information. Examples include (proprietary) electronic "Business Cards" available with certain brands of mobile phone, the popular vCard [3] format from IETF (also used for business card data), proprietary formats for appointments (e.g. the format used in MS Outlook®) and the IETF iCalendar [4] format for calendars (also known as iCal). Other formats have been proposed by the Microformats community [5] which is attempting to develop a set of simple open specifications for specific classes of data (e.g. contact info, calendar info, reviews…). To date however, relatively few mobile applications exploit the potential of these formats to facilitate transfer and usage of information between mobile users.

1.2 Our contribution and results of the project

The S.M.S. project has designed the generic MEM format and the specific format for different types, or "classes" of MEMs. Moreover the project designed and implemented: 1) a library for using the MEMs; 2) a framework for exchanging MEMs (using the so-called SMILE middleware[2]); 3) a mobile client (called MOVE) that uses MEMs within a navigation application[3]. The library, the framework and the mobile client have been coded using the Java language. Java 2 SE is used for components running on PC/servers and J2ME (MIDP 2.0, CLDC 1.1) for the mobile client, which can run on the large number of symbian smart-phones and windows mobile phones that support a Java virtual machine. The applications and the code are available at [6] under an open source licence, which could facilitate the spreading of the MEM concept beyond the scope and the duration of the S.M.S. project. The project is currently running a field trial in a University campus, involving a large group of students [7].

2. MEMs

In our vision, a MEM is an electronic "note", containing a structured set of attributes associated with a specific class of information (e.g. information describing a location, a person, a service, or a Web site). Users can capture, annotate and store MEMs associated with their current environment (e.g. a business card for the restaurant where they eating or the person they are talking to) or produced by a service they are using (e.g. a confirmation of booking from an airline service). MEMs are readable both by humans and by computer applications: users can browse their MEMs for useful information or pass them to MEM-enabled applications which use the embedded information to provide special services (e.g. purchasing a ticket for a show described in the memo).

The software developed by the S.M.S. project makes it easy for users to send MEMs to other users, to share them with a broader community, or to use them as input for online services. Thus a user in a specific location can capture a MEM for the location and send it to a friend. The friend can pass the MEM to a navigation tool which will guide her to where her friend is waiting. It is possible to embed MEMs in email, or even in Instant Messages or SMS (reducing the carried information). These adaptations make the MEM accessible to users who do not have special software. MEMs can be made available on the web and easily downloaded by mobile users.

A set of additional tools make it easy for service providers to offer support for users from different countries speaking different languages. Cryptographic mechanisms guarantee user privacy. The authenticity of MEMs of a sensitive nature (e.g. those containing details of a financial transaction) is guaranteed by digital signatures.

2.1 The structure of a MEM – the end-user view

From the end-user point of view, a MEM appears as an icon which can be expanded into a "card" showing a certain number of fields associated with a specific context. When MEMs are transmitted via email, IM, SMS or other messaging systems, they may appear as URLs.

Different classes of information (e.g. information about people, locations, web sites, services) have different classes of MEM. Some of these have been pre-defined, other specific classes can be created by service providers (e.g. an airline could define a special class of MEM for "flights"). Users and service providers have the possibility to expand pre-existing MEMs by adding new fields.

---

[2] SMILE (Simple Middleware Independent LayEr) is an abstraction layer between the application and the underlying middleware platform, provided as a set of JAVA APIs. SMILE-JS (Json over SIP) is a "binding" of SMILE that runs on a broad range of Mobile Devices (J2ME MIDP2.0/CLDC1.1). Coding of messages is based on JavaScript Object Notation (JSON). Messages are transported by using the SIP protocol.

[3] The MOVE client is a browser for mobile services running on mobile terminals. MOVE allows users to instantaneously access services that match their preferences and current context. Other features include outdoor and indoor navigation, MEM management, information services and (on Windows Mobile phones) the ability to directly interact with the SIM. MOVE runs on Java 2 Micro Edition (CLDC) on Symbian and Windows Mobile phones.

2.2 Creating, publishing, capturing and sharing MEMs

Users and service providers will be able to use a range of different tools for creating and publishing MEMs. In the simplest case, users choose a template from a Web site (e.g. a template for a restaurant MEM), adapt the content and layout to their needs, define the "context" where the MEM will be available (lifetime, privacy level, locations where the MEM can be captured by other users) and "instantiate" the MEM, clicking on a "publish" button. More sophisticated users may design their own MEMs using tools similar to web editors.

S.M.S. envisions a number of ways in which a user can capture a MEM:

- Location-based capture: the S.M.S. system identifies and lists MEMs associated with the user's current location (e.g. a restaurant). To capture a MEM the user simply clicks on it
- Physical capture via near field transmission: S.M.S. lists MEMs associated with near field transmission devices in the vicinity of the user (e.g. a business card associated with the user she is talking to)
- Optical capture: a code representing the URL for the MEM (e.g. a bar code) is embedded in a physical artefact (e.g. a piece of publicity, a Powerpoint presentation). The user points his terminal at the code, and captures the MEM.

Once a MEM has been captured it can be annotated (e.g. the user can add comments) and saved in local or network storage. Users can retrieve the MEM at a later date, view it in human readable format, or pass it to an application or service. Examples of such services might include:

- buy/rent a product or an item described in the MEM
- navigation services which guide the user to a location embedded in the MEM
- map services, which geo-references the MEM onto a map
- reminder services, which uses the MEM to notify the user whenever a given condition is met (e.g. a given time has arrived or the user is close to a location).

A key feature of MEMs is that they are easy to share and send to other users. Two kinds of interactions are allowed:

- Creation and Exchange of MEM between peers. Physically, this interaction can be direct, or mediated by a server (in case the intended recipient is not online when the sender sends the MEM). Alternatively they can embed the URL for the MEM in emails, IM or SMS messages.
- Creation and Storage of a MEM in a repository ("MEM Server"). MEMs intended for use within a specific community or for users at large can be published to "MEM servers", which make them available to end-users in the relevant context.

2.3 Security and privacy

Since MEMs may contain sensitive personal data (e.g. identity data, credit card numbers, data on users' personal habits, contacts etc.). Protecting this data is of key importance for public acceptance. By default, requests for MEMs are anonymous or pseudonymous. Personal information is included in the request (and in MEMs themselves) only when it is essential to the purpose for which the MEM is being used. MEMs may be digitally signed, thus allowing service providers to prove their identity to end-users through common identification mechanism adopted in the Web. S.M.S. showcased that MEMs may also be signed by end users, in a secure way, directly from their mobile phones using secret keys contained in their USIM. MEMs implement non-repudiation capability and are suitable for proof of purchases and subscriptions. Finally, MEMs sent to a specific individual (or group of individuals) may be encrypted, both during transmission, and when they are stored (on the local terminal or on the network).

3. Design and implementation

In this Section we briefly describe the design and our reference implementation of MEMs (more details can be found in [9]). A MEM is a sequence of “fields”. A “field” in turn can be a “structure” with several subfields. When deciding upon the representation format of a MEM, we compared the options of using XML and JSON [10]. We decided to use JSON as more lightweight in terms of the needed processing and length of the text representation, thus better suited for our mobile device implementation. However, we kept a conceptual compatibility with XML, so that it is possible to serialize a MEM in JSON or XML. As XML defines the concept of XML schema (XSD), XSD might be used to syntactically describe the structure of a MEM. Thus, MEM schemas may be defined using XML schema definition and XML namespaces. Some examples are reported in [9].

3.1 General structure

We assign to MEMs the new MIME media type `application/vnd.ist-sms.mem+json`. A MEM contains a single field with meta information, with the reserved field name "META", followed by an arbitrary number of fields.

```
{ "META":{ "metafield1": "value1",
           "metafield2": "value2"
         },
  "BODY":{ "field1":"value1",
           "field2":"value2"
         },
  "ENCL":[ {"content1":{...}},
           {"content2":{...}}
         ]
}
```

Figure 1: Overall structure of a MEM (JSON notation)

Contents within the MEM are grouped within a single “BODY” tag. Inside this tag, fields can refer to other data, e.g. pictures, WEB information, IDs, or even other MEMs. Such data can be simply referenced through their URI or they can be actually embedded in the MEMs.

The reserved field name "ENCL" (for “enclosure”) is used to attach an arbitrary content to a MEM. The content of the “ENCL” field is an array, thus more than one object can be attached to a MEM. To refer to an object contained in or referred from a specific enclosure within a MEM, we use the XPath-like syntax “\ENCL[i]", where the index i is used to point to the target enclosure. Contents inside the “BODY” tag may be signed and/or encrypted.
Many examples of MEM fields are given in [9], showing how MEMs can support multiple fields of the same type (e.g. multiple “comment” fields), pictures and other binary objects and subfields inside MEM fields.

3.2 Meta data information in MEMs

In the following we briefly list some basic MEM meta data that we defined up to now. Further examples and explanations can be found in [9].

The “format” field identifies the MEM schema upon which the MEM is based. MEMs can be specialized starting from more generic MEM schemas, thus this field might be an array, specifying a number of formats to which the MEM conforms to. Formats must be listed from the most specific to the most generic. An application that processes MEMs should try to interpret the MEM according to the most specific format it is able to handle. We report an example of MEM in Figure 2.

The "serial" field uniquely identifies the MEM. It is assigned by the authoring application. The "author" field identifies the author of the MEM in a human readable way, whereas "authorID" identifies the author of the MEM in a "machine readable and addressable" way. This field should contain a valid URI (e.g. a SIP URI, a TEL URI, an email address or a web URI) identifying a digital subject. The "factory" field contains an URI that uniquely identifies a "resource" that has produced the MEM, and helps machines to correlate MEMs that relate to the same information. For example, consider MEMs containing highly volatile information, e.g. stock quotes (or scores for a football match); all MEMs containing information for the same stock and produced by the same authoring application are assigned the same value for the "factory" field. The "creationStamp" field records the creation date and (optionally) the creation time of the MEM. The "sender" field identifies the sender of the MEM in a human readable way; if the MEM is forwarded, the sender will be different from the author. The "senderID" field identifies the sender of the MEM in a machine readable way, allowing obtaining more attributes (like public keys) in a non ambiguous way. The "heuristics" field allows or discourages an application to parse the MEM heuristically. The "expiration" field informs that the information contained in the MEM will not be valid anymore after a given date. The "privacy level" field defines a privacy level for the MEM. MEMs with privacy level "public" are publicly available to all people and are called "public MEMs".


```
{
   "META" :
     {"format": [http://ist-sms.org/MEMs/restaurant,http://ist-
     sms.org/MEMs/place]
      "serial": "http://rpstr.example.com/2007-0107_123345/3ABC2342F3",
      "author": "Joe Downson",
      "authorID": "jdowson@iptel.example.com"
      "creationStamp": "2006-11-05T11:43+1",
      "sender": "Stephen Smith"
      "senderID": "ssmith@iptel.example.com"
      "heuristics" : false
     },
   "BODY":
     {"name": "The Bavarian Restaurant",
      "address":"Residenzstrasse, 12",
      "city":"Munich",
      "country":"Germany",
      "geoPos" : {"lat" : "48.139931" , "lng" : "11.577573"},
      "openingTimes" : {"openingDays":"from Tuesday to Sunday",
                        "openingHours":  "11.30 - 15.00/19.30 - 24.00"
                      },
      "telephoneNumber":[{"": "+ 49 089 3758 376"},
                         { "": "+ 49 089 3748 268", "comment": "Use on
     closing hours"}
     . . .
}
```


Figure 2: Example of a "restaurant" MEM

3.3 MEM schemas and processing of MEMs by applications

MEM schemas provide a "common language" to define the content of a MEM of a given type. An application is not obliged to fully validate the MEM before interpreting its content. Validation includes checking the content of a field, e.g. integer in the corresponding XSD schema, but strings in the actual serialized form of the MEM. Specific serialization format syntax checking, (i.e. well-formedness tests for XML) should, however, be applied; i.e.

improper balanced brackets and/or not matching XML tag closures should inhibit further processing. The processing application is allowed to interpret the MEM without using any knowledge of its schema, i.e. using a common meaning of the tags as English nouns or verbs, unless heuristic processing is disabled by setting the “heuristics” field to false in processed MEM meta data.

An application processing a MEM may “semantically” process some common fields defined for the MEM according to the relevant MEM schema. For example in a restaurant MEM the “name” field will identify the restaurant name and the “address” field will identify the address of the restaurant. The application that displays the MEM could translate these field names into corresponding names in the language of the user. A MEM can also contain information that the application is not able to process semantically (i.e. fields that are not included in the MEM schema). In this case the application may simply display the information (field name and field value), or just ignore them, as indicated by user preferences.

3.4 MEM classes

MEMs are very flexible and there may be an unlimited number of different types of MEM; however, we have found useful to identify four basic classes of MEMs to which each MEM type can refer to. A MEM class defines the minimum set of fields that a MEM of that class has to contain and specifies the minimum set of requirements a MEM-enabled application should implement to parse those MEMs. We defined the following classes:

- **Place:** MEMs contain information about public places, private places like houses and flats, shops and activities, generic locations;
- **People:** MEMs used to describe people (human beings or characters);
- **Events:** MEMs representing an activity held at a given time (e.g., football matches, lecturea in a classroom, parties, weddings, beer festivals, etc.);
- **Physical Entities:** MEMs identifying entities which are not people, like animals, paintings, cars, etc.
- **Memo:** MEMs containing information that cannot be classified into the previous four classes. For example a “todo” list, a list of person to be contacted.

These classes and their schemas are fully defined in [9]. It is also possible to define MEMs for specific contexts, as special cases of the above classes, e.g., “Monument” MEMs and “Restaurant” MEMs are special cases of MEM belonging to the class “Place”.
In any case, service providers and users can easily extend the original templates by adding new fields (marked by personalized “tags”). We expect that when other users recognize the usefulness of a tag they will start using it themselves. In brief, the MEMs extensible format creates rich opportunities for ‘emergent semantics’ [8].

Finally, Figure 3 reports some screenshots showing how MEMs are managed on a mobile phone by our MOVE application.

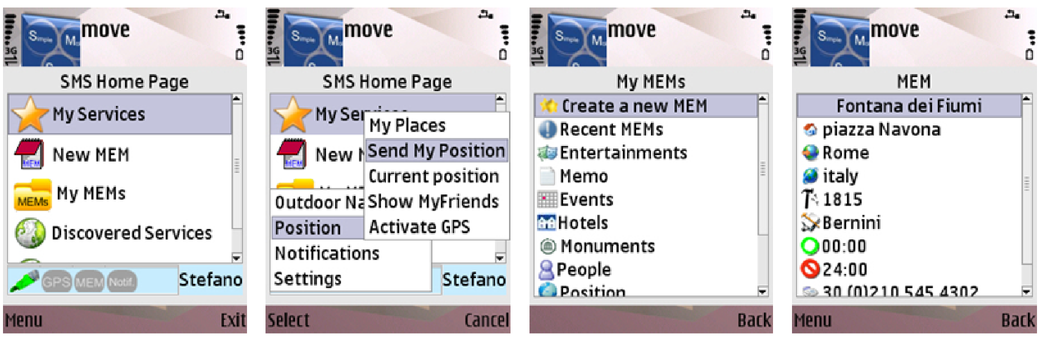

Figure 3: screenshots from the MOVE application

3.5 Security aspects

Security aspects are a fundamental concern in the sharing of information through MEMs. The detailed specification of the signature/encryption mechanisms used in S.M.S. are out of the scope of this paper. Here we give only a sketch of a solution defined by the project. A "SIGN" field in a MEM allows providing an hash or a signature of the MEM content, the whole MEM or a part of it. The "SIGN" field can be added at whatever level in the MEM structure. The value corresponding to the "SIGN" field is an object that can carry a hash or a digital signature. The hash or signature extends to all the fields at the same level of the "SIGN" field except the "SIGN" field itself.

3.6 Commenting, tagging and rating MEMs

In order to build a mobile social community, the capability to comment, tag and rate MEMs is very important. Comments by the original creator of a MEM should not be confused with comments coming from other users that received and used the MEM. Hence the need of: i) a message that can transport a specific MEM (or a reference to it) and a comment/tag list/vote related to the MEM and expressed by a user. This message can be send from a user to a user or from a user to a server; ii) a message carrying a given MEM and a set of comments/tags/votes that have been assigned by users to this MEM. Typically this kind of MEMs are public MEMs, available to all users for browsing, stored in a server and sent to a user upon user's request. For example, a "what's here" functionality, requesting all MEMs related to a given location, can be implemented into a MEM server.

4. Conclusions – MEMs as an extended clipboard

In this paper, we have briefly illustrated the concept of MEMs, as developed within the EU S.M.S. project. It is important to note that MEMs, together with supporting middleware, server-side components and the MOVE client is a complete and working solution, able to operate on most commercial devices, open source and ready to be exploited in real life [6][7]. We believe that this solution can play an important role in encouraging the take-up of mobile services by private and business users.

5. References

[1] Simple Mobile Services (S.M.S.), IST project 2006 034620, http://www.ist-sms.org/

[2] G. Bartolomeo, N. Blefari Melazzi, G. Cortese, A. Friday, G. Prezerakos, S. Salsano and R. Walker, "S.M.S.: Simplifying Mobile Services – for Users and Service Providers", International Conference on Internet and Web Applications and Services, ICIW'06, February 23-25, 2006, Guadeloupe, French Caribbean, published by IEEE Computer Society Press

[3] F. Dawson, T. Howes, "vCard MIME Directory Profile", IETF RFC 2426

[4] F. Dawson, D. Stenerson, Iinternet Calendaring and Scheduling Core Object Specification (iCalendar)", IETF RFC 2445

[5] The Microformats community, see http://microformats.org/, http://microformats.org/about/, http://microformats.org/wiki/

[6] Simple Mobile Services platform: http://netgroup.uniroma2.it/SmsPlatform/

[7] Simple Mobile Services community portal: http://netgroup.uniroma2.it/SMS-community/

[8] K. Aberer et al., 2004, "Emergent semantics, principles and issues", Lecture Notes in Computer Science, Springer Verlag.

[9] S. Salsano, G. Bartolomeo, R. Glaschick, R. Walker, N. Blefari-Melazzi, "Mobile Electronic Memos (MEMs) for Simple Mobile Services",
available at http://netgroup.uniroma2.it/twiki/pub/SMS/TechnicalReports/tr-MEMs.pdf

[10] "Introducing JSON" http://www.json.org/