Sharing Mobile User Experiences with Context-Based Mashups

Luca Costabello [*]
Politecnico di Torino
Corso Duca degli Abruzzi, 24
Turin, Italy
luca.costabello@studenti.polito.it

Oscar Rodriguez Rocha
Telecom Italia Lab
Via Reiss Romoli 274
Turin, Italy
oscar.rodriguezrocha@guest.telecomitalia.it

Laurent Walter Goix
Telecom Italia Lab
Via Reiss Romoli 274
Turin, Italy
laurentwalter.goix@telecomitalia.it

ABSTRACT

We describe two mashups based on context data and multimedia content retrieved from a client application running on mobile devices. This information is processed by our Context Aware Platform that provides specific APIs to deliver context data to our mashups.

First of all, these APIs enable context tagging of pictures and videos acquired by the users. Machine tags are adopted in order to enable context-filtered representation of pictures and videos on our content management system.

Moreover, platform APIs are used to gather context information that is analyzed by ad-hoc clustering algorithms to detect the users' daily actions. We therefore automatically generate structured blog posts that describe the users' day. The same information is represented as video blogs, too.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Clustering; H.3.5 [**Online Information Services**]: Data sharing, Web-based services

General Terms

Design, Algorithms, Experimentation

Keywords

Context awareness, cluster analysis, mashups, tagging, blog, user generated content

1. INTRODUCTION

The widespread popularity of mashups is one of the key features of the contemporary web. User-generated content and context information play a crucial role in this scenario. We face the issue of organizing the data collected from mobile users by our platform, in order to deliver context aware mashups aimed at easing the sharing of context between users. We describe two mashups that use this info: our semantic-tagging enriched content management system and the Automatic Blog Generator of daily users' actions.

[*]The work presented in this paper was done while the author was an intern at Telecom Italia.

Our work focuses mainly on user location and on nearby buddies. We work with many location technologies, such as Bluetooth and GPS, but most of our position data is cell-based. We concentrate on cell-based location, since the adoption of GPS devices is still not so widespread.

2. SYSTEM ARCHITECTURE

Figure 1 describes how our mashups hook up to the context awareness platform.

The client application running on mobile devices provides both raw context data and multimedia content. Raw context information, such as cell IDs, is delivered to the Context Awareness Platform [5], while pictures and videos are sent to the Contextual Tagging module (Figure 1).

Raw context data collected by clients is processed by the platform (e.g. cell IDs are resolved into civil addresses). The Context History module stores all the context updates performed by the users. RESTful APIs provides this enriched context information to our mashups.

When users take a picture, the client application prompts for custom tags. After this step, the content and the associated timestamp is uploaded to the Contextual Tagging module. Meanwhile, context updates are sent to the CA Platform. This raw context information is processed by the CA Platform and published on the content management system. This allows content filtering and browsing according to different context parameters. Content sets can be created according to a specific context (Section 3). Feeds are used to share information with other applications.

The blog generator mashup gathers users' daily context data via the platform APIs and carries out the task of creating and publishing users journals. Detailed personal blog posts describing user days are therefore published without the need for user intervention. Data mining is needed in order to infer high level information from users' location data. Ad-hoc clustering algorithms detect the users' daily actions, that are eventually converted into natural text and enriched with multimedia content, as explained in Section 4.3.

3. SHARING CONTEXTUALIZED MULTIMEDIA CONTENT

User-generated content management is our first mashup of context information. The following steps need to be performed:

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2008.3978
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2008.3978

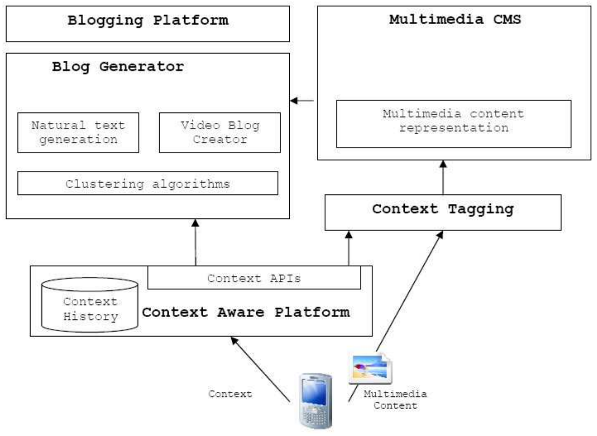


Figure 1: System architecture

3.1 Content and context upload

When the users take a picture or a video with their smartphones, our client application prompts for custom tags and for a title. The picture, the user-defined tags, the title and the associated timestamp are uploaded to the Context Tagging Module.

To overcome problems of limited connectivity and battery management, the client supports a deferred content upload procedure. Pictures and videos are associated to their creation timestamp (the Tagging Module will therefore add the right context information to the content).

Moreover, the client periodically provides the platform with context data, even if no picture needs to be uploaded. Cell IDs, nearby Bluetooth devices, GPS location are therefore sent to the platform with an user-defined frequency [5].

3.2 Content tagging

After being uploaded, each content is processed by the Tagging Module, which adds the user's context tags.

Context data is retrieved from the CA Platform, that provides the location, the nearby buddies and the calendar entries associated to the moment in which the picture has been taken. Focus is given on location: our platform converts GPS coordinates or Cell IDs into civil addresses. Moreover, it handles the retrieval of user-defined location labels (*Virtual Places*).

Context tags are generated according to the Machine tags specification [7]. We suggested brand new namespaces (`address`, `people`) that we use in addition to `cell` and `geo` namespaces (widely used by ZoneTag and Flickr). In addition, the `place` namespace is used with the predicate `is` and the value `crowded` (`place:is=crowded`). This tag is added to pictures taken when the user was near many visible Bluetooth devices, and therefore possibly located among many people. Figure 3.a shows a machinetagged Picture published on our portal.

3.3 Context-Based Content Management System Features

Machine tagged pictures and videos can be dynamically displayed as tag-based collections: it is therefore possible to filter pictures by Machine tag namespaces, predicates or values. For instance, it is possible to show only the pictures taken by a specific user (e.g. `people:fn=Walter+Goix`), pictures taken in a cell (e.g. `cell:cgi=460-0-9522-3661`), content acquired in a crowded area (`place:is=crowded`), etc...

Furthermore, content navigation is improved by the dynamic generation of clouds for Machine Tag namespaces, predicates or values. It is therefore possible to create context based tag clouds (e.g. a cloud for all the cities, for a specific person or a civil address, etc...).

Machine Tags are displayed in a friendly format and are therefore separated from user-defined tags. For example, `address:city=Torino` is displayed on screen as: `Where?In Torino` (Figure 3.a).

As seen in Figure 3.a, a map is embedded in the page. Our platform provides the location error (this is particularly useful for cell-based context data, where the error is the cell radius). The uncertainty is therefore plotted on the image.

4. EXPLOITING CONTEXT FOR BLOGGING

Another mashup that relies on context information and on user-generated content is our blog generator. We automatically generate structured blog posts that describe the subjects' days. Context data is processed by our clustering algorithms before being converted into natural text and published on the web.

4.1 Context data issues

Cell-based location data follows different patterns according to *where* the user is located. Base stations concentration is higher inside urban areas, and cells have a few hundred meters span. This situation allows a more accurate user location. On the other hand, inside a dense populated area, base stations overlap and clients may experience frequent cell handovers, even if the user is not moving. Detecting a static situation is, in this case, a more complex task. Nevertheless, detecting fast movement is easier (e.g. user driving). A slow moving subject (e.g. a passer by) generates patterns that can be confused with a static situation affected by many cell switches.

Rural areas are typically covered by a small number of base stations, spanning a few Kilometers. Location is therefore less fine-grained. In these circumstances, static actions are easier to detect, since there are fewer overlapping cells. On the other hand, intra-cell movements cannot be detected, since no cell handover is performed.

4.2 Clustering algorithms

An average day is typically made of hundreds of context records. Our aim is to detect what kind of *actions* the users have done during the day (Figure 2.a).

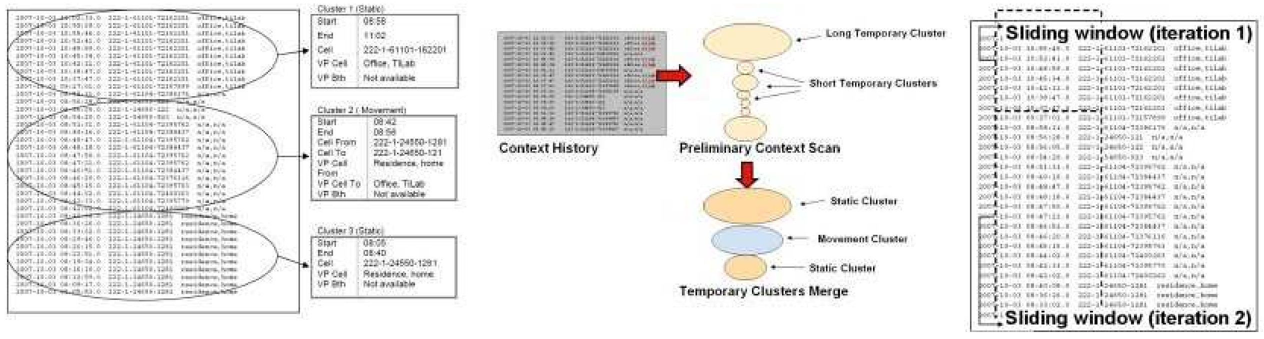
Clustering process works on location and time dimensions. Both cell ID and Virtual Place labels are strings. Distance-based clustering algorithms are therefore not suitable for the task, since it is not possible to define the euclidean distance for cell IDs or user-defined labels. Furthermore, we have to deal with time, to respect the chronological order of events [3].

The context-based blog generator detects both static and movement situations.

Two algorithms have been designed, in order to work with different data patterns.

Compare&Merge algorithm works as follows (Figure 2.b) [3]. Chronologically near records with the same loca-

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2008.3978
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2008.3978



(a) (b) (c)

Figure 2: (a) Context data cluster analysis goal. (b) Compare&Merge. (c) Multi Level Sliding Window

tions are merged. This step detects non-ambiguous static situations, i.e. time spans during which the user has performed several context update with the same location (e.g. the same cell ID or the same Bluetooth/cell ID user label). Our trials show that this is one of the most common situation, so it has to be detected properly. At the end of the scan we get a list made of *long temporary clusters*, each of them clearly marking a static situation, and *short temporary clusters*. The latter can be part of a movement situation, but the user could have been affected by spurious cell handovers, instead (Section 4.1).

The temporary cluster list is therefore processed. Particular attention is given on short temporary clusters: these are merged in a wider cluster matching the original chronological order. Before being added to the final list, this newly created cluster is analyzed. If it includes too many different locations (a proper threshold must be set), it is labeled as a movement cluster, otherwise it is a static one. In this case, the most frequent location included in the cluster must be set as the cluster center.

Compare and Merge is challenged by long-lasting, non-labeled context data sequences affected by frequent cell handovers. These patterns might be confused with movement actions, even if the user is not moving. The problem can be lessened by marking the switching cells with the same user-defined labels, but the workaround cannot obviously be used in all cases (users cannot label every visited place).

MultiLevel Sliding Window algorithm solves the issue, without changing input data format (Figure 2.c) [3]. Clustering is performed in a hierarchical way starting from user labels, both Bluetooth or cell based. If there is no labeling information, the algorithm works on cell IDs. Context data is processed in chronological order, to comply with the actions' chronology. A sliding window is used to process each user's context history. The time window has a fixed *maximum* length, expressed in minutes by a specific parameter.

The algorithm works therefore in a time related fashion, without being aware of how many records are included inside a window in every iteration. Context data records occur with non deterministic frequency [5]: tuning window length on record number would have been incorrect. Nevertheless, window duration does depend on user-defined context data update policies: the window must include enough records in order to obtain better results. If context updates occur with a low frequency, it will be better to extend the window duration.

Data can be merged in compliance with the defined location hierarchy (this is where the *MultiLevel* adjective comes from): the highest priority is given to Bluetooth Virtual Places, since they typically identify small areas (e.g. rooms, offices). If none of these is set, the algorithm works at the lower level (cell IDs Virtual Place). If no user label is set, the procedure works at cell ID level. However, the number of nested levels and the related data bindings can be edited, in order to use the algorithm in different scenarios.

4.3 Structured blog post creation

Having performed cluster analysis, merged data has to be converted into natural text, in order to create the users' blog post.

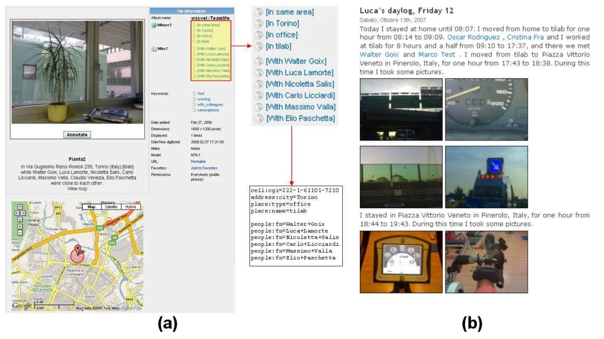
Text Generation. Standard sentence structures have to be defined (e.g. sentences for both static and dynamic situations) and suitable verbs must be chosen too, according to the presence of movement or to what type of Virtual Place is set (e.g. *to work* if the user has been in his office, *to shop* for a mall or another shopping area, *to move* for dynamic clusters, etc). Text information can be customized too, according to the desired detail level (e.g. nearby users can be showed or not, action durations can be hidden, etc...).

Microformats. Many sentence blocks such as civil addresses and nearby people are enriched by the use of Microformats semantic markup techniques [2]. We use `adr`, `XFN` and `hCard` Microformats, formatted in order to preserve the structure of sentences.

Embedding content. Blog posts are enriched with contextualized multimedia content. Pictures (and movies) taken during the day are embedded inside text and are assigned to the context in which they have been acquired. Multimedia files are retrieved via RSS feeds from our portal (Section 3.2) and thumbnails are added inside text.

4.4 Context Aware VideoBlog

A further module of our blog post generator converts clustering algorithms' output into personal daily video blogs. User's actions are listed in a headline news fashion: the animation is a mix of user context data and multimedia content acquired during the day. Pictures and videos are retrieved from our portal, via RSS feeds. Videoblogs are created by the open source PHP library, Ming [6]. Data is transferred


File information
Album name: wasvel/Teamlife
Where?: [In same area] [In Torino] [In office] [In tilab]
Who?: [With Walter Goix] [With Luca Lamorte] [With Nicoletta Salis] [With Carlo Licciardi] [With Massimo Valla] [With Elio Paschetta]
Keywords: Test working with_colleagues cameraphone
Annotate
Map
Satellite
Hybrid
(a)
(b)

Date added: Feb 27, 2008
Dimensions: 1600 x 1200 pixels
Displayed: 1 times
DateTime digitized: 2008:02:27 17:21:59
Make: Nokia
Model: N70-1
URL: Permalink
Favorites: Add to Favorites
Permissions: Everybody (public picture)

Pianta2
In Via Guglielmo Reiss Romoli 239, Torino (Italy) [tilab] while Walter Goix, Luca Lamorte, Nicoletta Salis, Carlo Licciardi, Massimo Valla, Claudio Venezia, Elio Paschetta were close to each other.
View map

```
cell:cgi=222-1-61101-7210
address:city=Torino
place:type=office
place:name=tilab

people:fn=Walter+Goix
people:fn=Luca+Lamorte
people:fn=Nicoletta+Salis
people:fn=Carlo+Licciardi
people:fn=Massimo+Valla
people:fn=Elio+Paschetta
```

Luca's daylog, Friday 12
Sabato, Ottobre 13th, 2007
Today I stayed at home until 08:07. I moved from home to tilab for one hour from 08:14 to 09:09. Oscar Rodriguez , Cristina Fra and I worked at tilab for 8 hours and a half from 09:10 to 17:37, and there we met Walter Goix and Marco Test . I moved from tilab to Piazza Vittorio Veneto in Pinerolo, Italy, for one hour from 17:43 to 18:38. During this time I took some pictures.

I stayed in Piazza Vittorio Veneto in Pinerolo, Italy, for one hour from 18:44 to 19:43. During this time I took some pictures.

Figure 3: A machine tagged picture (a) and a sample blog post (b).

from the clustering module using an ad hoc XML format.

4.5 Running the blog generator

The blog generator is implemented as a standalone application. The program runs in batch mode, and it is scheduled daily at 5:00 am. User days are assumed to start at the same moment. At this time most users are still asleep and, on the other hand, night actions after midnight can be included in the previous day, even if they are part of a new one (most subjects still perceive them as part of the previous day).

5. RESULTS

Our content management system has been tested for several months, hosting nearly 150 CA platform users. Until now, more than 9,000 multimedia content have been published. Nearly all of these pictures and videos have been automatically machinetagged.

A smaller group of ten users has been testing the blog generator, too. This trial phase has been running for over nine months. The use of clustering techniques reconstructs what the users did during each day, trying to deliver an output similar to their memories [3].

6. RELATED WORK

The term *Machine tag* has been proposed by Flickr [7]. Flickr publishes these metadata separated from simple, user defined tags. Querying Machine tags is also possible by using ad hoc APIs, but no graphical Machine tag navigation or interpretation is provided.

Basically, our client has the same purpose of other well-known applications, such as Yahoo! Research ZoneTag [10] [1], Shozu [9] or Nokia Location Tagger [8]. Our solution can work with both GPS and cell-based location (Nokia Location Tagger uses GPS data only). Our platform supports cell ID reverse geocoding. Unlike ZoneTag, being a network operator allows us to do this with no need for users intervention (this feature is only available on our Italian mobile network).

Other works focused on automatic generation of user action logs have been carried out. Mobilife Life Blogging [4] fetches context data from user devices (PDAs, smartphones), from which a client gathers raw information. Context data is used to create blog posts. Merging algorithms are used, but time dimension is not considered.

7. CONCLUSIONS AND FUTURE WORK

Our content management system is crucial to the development of other CA Platform-based mashups. Context-based Machine tags navigation and filtering improves content retrieval and browsing, and increases the user experience (Section 3.3). Multimedia Content can be uploaded to the Context Tagging Module via MMS, too. However, clientless usage is still under development.

The blog generator's clustering algorithms let us group and merge user context data, with special concern to cell ID location. To achieve this goal, context data issues have to be faced (Section 4.1). Two algorithms have been designed, in order to fit different context data patterns (Section 4.2). Trials with real life information helped tuning the algorithms' parameters in order to minimize the error. Clustering process and natural text generation eased the analysis of how many daily activities details the users remember [3]. This facts have been exploited to design a blog post text structure and to choose the best detail level (Section 4.3). Besides, Microformats are used to embed location data. Privacy is a crucial issue to deal with. We still do not include a privacy level selector for each user's blog posts. The feature will allow users to choose who can read their daily logs (e.g. private, friends only, public) and which kind of information display (Section 4.3). Blog posts are now isolated. Future developments will allow new posts to be aware of previous days history, in order to deliver a better user experience.

8. REFERENCES

[1] S. Ahern, M. Davis, D. Eckles, S. King, M. Naaman, R. Nair, Mirjana, Spasojevic, and J. H.-I. Yang. Zonetag: Designing context-aware mobile media capture to increase participation. *Workshop on Pervasive Image Capture and Sharing, Ubicomp06*, 2006.

[2] J. Allsopp. *Microformats: empowering your markup fow Web 2.0.* Friends of ED, 2007.

[3] L. Costabello and L.-W. Goix. Time based context cluster analysis for automatic blog generation. *WWW2008 Workshop on Social Web Search and Mining*, 2008.

[4] J. Koolwaaij, A. Tarlano, M. Luther, P. Nurmi, B. Mrohs, A. Battestini, and R.Vaidya. Context watcher - sharing context information in everyday life. In *Proceedings of WTAS*, 2006.

[5] L. Lamorte, C. Licciardi, M. Marengo, A. Salmeri, P. Mohr, G. Raffa, L. Roffia, M. Pettinari, and T. S. Cinotti. A platform for enabling context aware telecommunication services. *Third Workshop on Context Awareness for Proactive Systems*, 2007.

[6] Ming, a swf output library and php module. http://ming.sourceforge.net.

[7] Machine tags. http://www.flickr.com/groups/api/discuss/72157594497877875/.

[8] Nokia location tagger. http://www.nokia.com/betalabs/locationtagger.

[9] Shozu. http://www.shozu.com.

[10] Zonetag photos. http://zonetag.research.yahoo.com/.

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2008.3978
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2008.3978