A Proactive Application to Monitor Truck Fleets

Fábio da Costa Albuquerque, Marco A. Casanova
Department of Informatics
PUC-Rio
Rio de Janeiro, Brazil
{falbuquerque, casanova}@inf.puc-rio.br

Jose Antonio F. de Macedo
Department of Computing
Federal University of Ceará
Fortaleza, Brazil
jose.macedo@lia.ufc.br

Marcelo Tílio M. de Carvalho
TecGraf Institute
PUC-Rio
Rio de Janeiro, Brazil
tilio@tecgraf.puc-rio.br

Chiara Renso
Istituto di Scienza e Tecnologie dell'Informazione – CNR
Pisa, Italy
chiara.renso@isti.cnr.it

***Abstract*—Positioning systems, combined with inexpensive communication technologies, open interesting possibilities to implement real-time applications that monitor vehicles and support decision making. This paper first discusses basic requirements for proactive real-time monitoring applications. Then, it describes how to structure and geo-reference unstructured text information available on the Internet, with a focus on road conditions change and using available geocoding services. Lastly, the paper outlines an application that monitors a fleet of trucks and incorporates proactive features.**

Keywords—vehicle monitoring; proactive behavior; Twitter analysis.

I. INTRODUCTION

According to Tennenhouse [13], to be proactive, an application must detect interesting situations before they happen and must be able to handle such situations without human supervision.

In this paper, we start by listing the requirements for proactive applications that monitor moving objects. Then, we outline a proactive application that monitors truck fleets. The application uses workflows to model truck trips, monitors the location of the trucks and processes tweets to extract and geo-reference traffic-related facts about road conditions that may affect truck trips. It was deployed to monitor a medium-size truck fleet (about 500 trucks) operating for a liquid gas distribution company and a large truck fleet (about 5,000 trucks) operating for a gasoline distribution company.

The paper condenses work reported in [1][2][3]. However, the framework to process Twitter messages and the truck fleet monitoring application has not been reported elsewhere.

The paper is organized as follows. Section 2 describes the requirements for proactive monitoring applications. Section 3 discusses how to extract traffic-related facts from tweets. Section 4 introduces a proactive application to monitor truck fleets. Section 5 describes related work. Finally, Section 6 presents the conclusions.

II. PROACTIVE MONITORING OF MOVING OBJECTS

To achieve proactive behavior, an application that monitors moving objects must:

- Model the behavior of the moving objects and monitor the current state of the objects.
- Model the environment where the objects move and monitor the current state of the environment.
- Detect environment changes that may affect the future behavior of the moving objects and adjust their behavior to the changes.

A. Modeling moving objects behavior

We require that the application should include a model of the behavior of the moving objects and that the application should be able to monitor the current state of the objects.

If we regard a moving object M as the manifestation of a process P, modeling the behavior of M means to model process P. The accuracy of the model naturally depends on the requirements of the application. We assume that the application models process P as a workflow W_M.

The workflow language must support certain features to be useful for monitoring moving objects. We assume that each *step p* of W_M is associated with an *action* A_p, a *location* O_p and a *time interval* $[B_p,E_p]$, where B_p is the *beginning* of the interval and E_P is the *end*. Among the possible actions, we must include *moving* from O_p to a location D_p, called the *destination* of p, through a *route* R_p, within the time interval $[B_p,E_p]$; the route R_p must start on O_p and end on D_p.

Let W_M be the workflow associated with a moving object M. To monitor M, the system must be able to detect the current step of the execution of W_M, including the current location of M. All future decisions will be based on the current location and on the next steps of W_M.

Finally, the application must be able to assess whether the execution of the next steps of W_M is still feasible. Otherwise, an alternative workflow must be proposed in order to achieve the same effects of the remaining steps of W_M, considering the available routes and resources.

B. Modeling the Environment

We require that the application should include a model of the environment where objects move and that the application should be able to monitor the current state of the environment.

Typically, this means that the application must include a collection of data sources, classified as (geospatial) static data sources and dynamic data sources [1].

A *static data source* contains data about geospatial features with fixed location and low rate of update, such as roads and buildings. Hence, possible static data sources are (national) Spatial Data Infrastructures (*SDI*) nodes and Volunteered Geographic Information (*VGI*).

A *dynamic data source* delivers data about dynamic phenomena that temporarily modify static geospatial features. A dynamic data source is therefore essential to monitor the current state of the environment. As candidate dynamic data source, we may list Twitter channels, *RSS* and *geoRSS* feeds and Open GeoSMS [2].

C. Future Behavior of Moving Objects

We require that the application should be able to detect environment changes that may affect the future behavior of moving objects and that it should be able to adjust the behavior of the objects to the changes.

Recall from Section *A,* that we modeled the process associated with a moving object M as a workflow W_M. Therefore, when simulating the execution of W_M, the application may detect potential problems with the execution of future steps due to a change in the state of the moving object or in the state of the environment.

The application must then analyze data retrieved from both static and dynamic structured data sources (defining the state of the environment), data about the moving object (defining its state), and data associated with future steps to detect changes that may affect the current workflow execution. In particular, the system must be able to analyze the available data to detect facts that affect routes where the moving object may pass.

When a change in the environment affects future steps, the system must assess the feasibility of the workflow and suggest alternatives. The rules that govern this process are application-dependent. In general, the following are feasible alternatives: the workflow execution may be delayed until the environment returns to a state where execution can be retaken; the workflow may be replaced by another workflow that better fits the current environment state; the workflow may be adapted according to some predefined strategy.

III. Processing Traffic-related Twitter channels

As already mentioned in Section II.B, to adequately model environment changes, a proactive application must include dynamic data sources, such as Twitter channels.

To detect changes in the (traffic) environment, we propose to process Twitter channels broadcast by traffic authorities. The first step, described in Section III.A, is to structure the messages to extract facts that might impact road conditions. The second step, described in Section III.B, is to geo-reference the structured data to discover where the fact occurred.

A. Structuring Twitter Messages

We first note that Twitter messages from traffic authorities have some peculiar characteristics: they tend to be well structured and contain only idiomatic words; they may use hash tags and popular location names, which are dealt with a synonyms dictionary. These features facilitate processing such message streams, as opposed to Twitter messages generated by individual users.

Albuquerque et al. [2] describe an application that transforms unstructured traffic-related text messages from the City of Rio de Janeiro traffic authority (in Portuguese) into structured data that describe facts related to traffic conditions. The application is generic enough to be trained to similar Twitter channels (in other languages) and uses natural language processing and machine learning techniques. It has two phases: (a) the first phase uses a NER (Named Entity Recognizer) to discover relevant entities in the message; (b) the second phase creates a dependency relationship over the entities to produce structured data.

For example, using a corpus built from Twitter messages generated by the City of Rio de Janeiro traffic authority (in Portuguese), the application achieved an average f-measure for entity recognition of 90.41% and accuracy for information structuring of 73.37%.

B. Georeferencing Structured Data

The structured data that results from the first step, described above, contains information about facts of interest and their location, perhaps qualified with further information.

Such additional information is important to improve the quality of the location of a fact. Albuquerque et al. [3] describe an application that extracts the following attributes from the structured data that results for the first step:

Restriction attribute: defines the neighborhood where a fact occurred.

Reference attribute: indicates other locations near a fact location.

Direction attribute: indicates if a fact occurred in a two-way street, which is important to disambiguate the location of the fact (see Fig. 3).

IV. A Proactive Application to Monitor Truck Fleets

This section outlines an application that monitors a fleet of trucks and that incorporates proactive features. Briefly, the application has the following major modules:

Module 0: stores conventional data (about the truck fleet and customers) and geographic data (street/road maps and truck routes).

Module 1: obtains traffic-related messages from Twitter; transforms Twitter messages into structured, dynamic data; geo-references the structured data.

Module 2: monitors truck trajectories.

Module 3: analyzes the geo-referenced structured dynamic data to detect traffic-related facts of interest; determines which facts may affect planned truck trips; alerts the user about such facts.

To implement Module 1, we developed the Environment Web Monitor (EWM) framework, shown in Figure 1. The Scrapper component is responsible for retrieving unstructured Twitter messages. The NLP and Machine Learning components structure Twitter messages (see Section III.A). The Core component geo-references the facts (see Section III.B).

Module 2 processes the GPS data streams the trucks generate to detect certain types of events, that is, to extract trajectory semantics. Each record in a data stream contains the identification of the truck that sent the record, its location, time, and speed. The module detects the following events, among others (as illustrated in Fig. 2 – in Portuguese):

Speed Limit Exceeded: detected when a record indicates a speed above the road limit (for trucks).

Driving Time Exceeded: detected when a driver drives for over four hours without stopping for at least 30 minutes.

Stop at a Customer: inferred from a stop close to a customer. Each time a stop is detected, a spatial query is made, using the coordinates of the stop as the center of a circle with radius of 100 meters. The customer located closer to this center is associated with the stop.

Forbidden Area Entered: detected when a record indicates that the truck is inside a restricted area (i.e., regions or roads that do not allow truck circulation).

Module 3 detects if a fact might affect a truck trip and indicates the problem on the user interface (basically a map). As an example, consider the following (real) tweets:

(a) *"Acidente entre dois carros no Aterro do Flamengo".* (*"Accident between two cars at Aterro do Flamengo."*)

(b) *"Acidente envolvendo dois carros no Aterro do Flamengo, sentido #zonasul, na altura da Avenida Oswaldo Cruz."* (*"Accident involving two cars at Aterro do Flamengo, direction #zonasul, near Oswaldo Cruz Avenue."*).

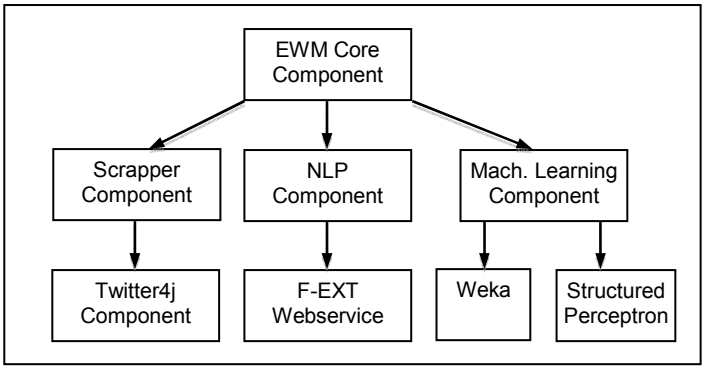

Fig. 1. Overview of the EWM components.

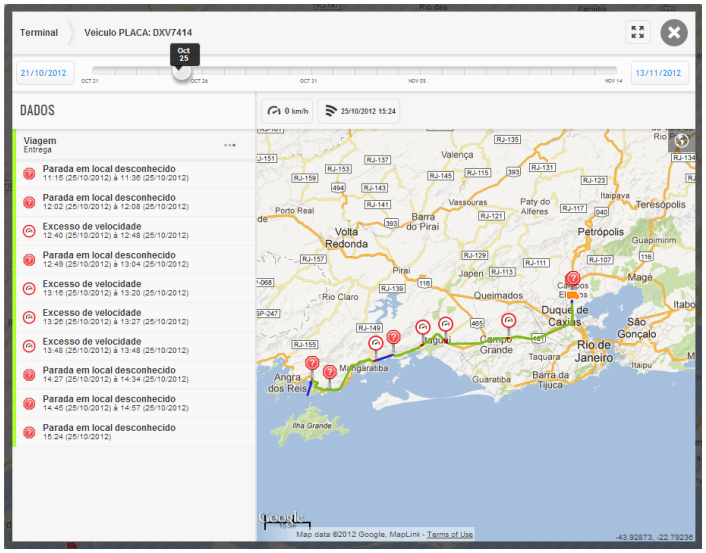

Fig. 2. Example of the user interface of the truck fleet monitoring application.

(a) (b)

Fig. 3. Examples of the user interface of the truck fleet monitoring application, depicting facts that affect truck trips.

Figure 3 illustrates the result of analyzing these tweets. The maps show the location and the type of facts of interest, such as traffic accidents, malfunctioning traffic lights, etc.

V. Related Work

The review of related work concentrates on two topics: achieving proactive behavior and processing Web data.

Achieving proactive behavior. Magoutas et al. [9] propose a proactive application that performs intelligent energy distribution to end-use customers, predicting the energy consumption based on historical data, information about the customers and events from various heterogeneous sources (e.g. social networks, customer`s GPS car system, etc). Santos and Moreira [4] investigate how to predict the next steps of moving objects based on their current location and road data.

The second approach to proactivity is based on the extraction of relevant facts that potentially affect the future behavior of moving objects. The architecture proposed by Fritz et al. [8] adopts an *a priori* crawling of the Web in search of relevant data about objects belonging to non-trivial categories, such as potential emergency shelters and hospitals.

Processing Web data. Sakaki et al. [10] discuss the semantic analysis of tweets. The authors explore, as use cases, the problem of estimating the location of an earthquake center and the problem of estimating the trajectory of a typhoon.

Earle et al. [7] also introduce a model for earthquake detection. Event detection is based on the increasing rate of earthquakes tweets. The location of a tweet is either retrieved from the tweet GPS data, or inferred from the static location string found in the user profile, using the Google Maps Geocoding Service.

As an example of processing RSS feeds, Chen et al. [6] use a rule-based tagger that extracts potential name places from a given text and matches them against a gazetteer. Carvalho et al. [5] classify tweets indicating traffic-related and non-traffic related facts, using machine-learning techniques.

Structuring raw text data and extracting relevant information is not a trivial task. The Locus system [12], an urban spatial finder, has an advanced search feature with a geo-referencing objective similar to ours, although with a different implementation. Borges et al. [4] use predefined patterns to extract addresses from Web pages using a set of regular expressions.

VI. Conclusions

In this paper, we first discussed basic requirements to achieve proactive monitoring of moving objects. Then, we outlined how to use Twitter message streams to detect changes in road conditions and to geo-reference the relevant facts. Finally, we briefly described a prototype application to monitor truck fleets. As mentioned in the introduction, the application has been deployed to monitor a medium-size truck fleet (about 500 trucks) operating for a liquid gas distribution company and a large truck fleet (about 5,000 trucks) operating for a gasoline distribution company.

A key point to achieve proactive behavior was to explore Twitter message streams. However, the techniques presented are not restricted to Twitter messages and can be extended to data sources of different natures. As future work, we intend to test the proposed approach with tweets from ordinary users. The *@LeiSecaRJ* channel, for instance, publishes a series of tweets with non-idiomatic words and excessive use of abbreviations.

Acknowledgements

This work was partly supported by CNPq, under grants 301497/2006-0, 475717/2011-2, by FAPERJ under grant E-26/103.070/2011, and by the European Community FP7-People-2011-IRSES SEEK Project.

References

[1] Albuquerque F. da C., Barbosa I., Casanova M. A., Carvalho M. T., Macedo J. A.: Proactive Monitoring of Moving Objects, Proc. ICEIS 2012, pp. 191-194.

[2] Albuquerque, F. da C., Bacelar F. C., Tapia X. A. C., Carvalho M. T.: Extrator de Fatos Relacionados ao Tráfego. Proc. SBBD 2012, São Paulo, Brazil, pp. 169-176.

[3] Albuquerque F. da C., Barbosa I., Casanova M. A., Carvalho M. T., Georeferencing Facts in Road Networks. Proc. GeoInfo 2012. pp. 120-127.

[4] Borges K. A. V., Laender A. H. F., Medeiros C. B., Davis C. A.: Discovering geographic locations in web pages using urban addresses. GIR 2007, pp. 31-36.

[5] Carvalho, S., Sarmento, L. and Rossetti, R. Real-Time Sensing of Traffic Information in Twitter Messages. Proc. 4th Workshop on Artificial Transportation Systems and Simulation, (2010).

[6] Chen, Y. F., Di Fabbrizio, G., Gibbon, D., Jora, S., Renger, B., Wei, B., Geotracker: geospatial and temporal RSS navigation. In WWW 2007, pp. 41-50.

[7] Earle, P., Bowden, D., Guy, M., Twitter earthquake detection: earthquake monitoring in a social world. In: Annals of Geophysics, 54(6) (2011).

[8] Fritz, C., Kirschner, C., Reker, D., Wisplinghoff, A., Paulheim, H., Probst, F., Geospatial Web Mining for Emergency Management (2010).

[9] Magoutas, B., Mentzas, G., Apostolou, D. Proactive Situation Management in the Future Internet: The Case of the Smart Power Grid. Proc. 22nd International Workshop on Database and Expert Systems Applications (2011), pp. 267-271.

[10] Sakaki, T., Okazaki, M., Matsuo, Y., 2010. Earthquake shakes Twitter users: real-time event detection by social sensors. Proc. WWW 2010.

[11] Santos, M. Y., and Moreira, A.. GUESS: on the prediction of mobile users' movement in space, In: Wachowicz, M. (Ed.) Movement-Aware Applications for Sustainable Mobility: Technologies and Approaches, IGI Global Publishing (2010).

[12] Souza L. A., Delboni T M., Borges K. A. V., Davis C. A., Laender A. H. F.: Locus: Um Localizador Espacial Urbano. Proc. GeoInfo 2004, pp. 467-478

[13] Tennenhouse, D. Proactive computing. Comm. of the ACM, vol. 43, (May. 2000), pp. 43–50.