A real time data streaming approach to best route planning

L.ROMANO*, L.COPPOLINO*, S.GIORDANO*, G.GUGLIARAβ
* Dipartimento per le Tecnologie
Università degli Studi di Napoli “Parthenope”
Centro Direzionale di Napoli – 80143 Napoli
ITALY
(lrrom, luigi.coppolino, savio.giordano)@uniparthenope.it

α Laboratorio ITeM "Carlo Savy''
Consorzio Interuniversitario Nazionale per l'Informatica (CINI)
Via Cinthia - Edificio 1, 80126 Napoli
ITALY

β Comune di Afragola
Piazza Municipio – 80021 Afragola(NA)
ITALY
g.gugliara@comune.afragola.na.it

Abstract: This paper describes our experience in building a framework designed for road conditions prediction. The proposed framework was made up by configuration and control panels based on web technologies, an engine for road conditions prediction based on stream processing paradigms, and a multimodal message delivery service which provides the user with multiple modes (SMS, MMS or email) of receiving update information from the system. The key design concept of the engine is the data monitor, which in real time keeps track of a sensor networks in order to correlate all the information available and to make the most accurate prediction of the road conditions. The framework has been implemented as a pilot application in Afragola, a small municipality in Napoli, a city in the South of Italy.

Key-Words: “Stream Processing” “Complex Event Processing” “Business Process Manager” “Grammar based parsing” “Quality of services” “SOA”

1 Introduction
This paper describes our experience in building a framework designed for road conditions prediction. The proposed framework was made up by configuration and control panels based on web technologies, an engine for road conditions prediction based on stream processing paradigms, and a multimodal message delivery service which provides the user with multiple modes (SMS, MMS or email) of receiving update information from the system. The key design concept of the engine is the data monitor, which in real time keeps track of a sensor networks in order to correlate all the information available and to make the most accurate prediction of the road conditions. The framework has been implemented as a pilot application in Afragola, a small municipality in Napoli, a city in the South of Italy.

The main goal of the framework is to provide the users with a service for the prediction of road conditions. A conceptual view of the framework is depicted in figure 1. The framework is composed by a Configuration Panel used by the system operators and by the users, sensor networks for data collection, an engine that compute the current best route, and a Message Delivery Service in charge of sending back to the users the current best route. In the following sections we will go through these three main part of the framework and in the final section we describe the pilot application.

2 The configuration panel
There are two kinds of actors in the configuration panel scenario, namely users and operators. The users through the configuration panel register themselves to the service. During the registration phase the users have to provide the
service both with personal information in order to create an account and with route preferences. Among the personal information is mandatory to specify either a phone number or an email address, because the framework will use this information to configure the Message Delivery Service according with the user preferences. With regard to the route preferences, the user has to specify for each route a starting point, a destination, and a default starting user location. Later the user can choose a favorite route among the routes proposed by the framework after he has filled in the previous fields. In figure 2 is depicted the UML use-case diagram which describes the operation available for a registered user.

In that diagram the “Check auth” functionality checks the user has the rights to perform all the other available operations. The system operators use the configuration panel for all the management operations among which the most important is the management of road condition events whose use case diagram is depicted in figure 3. When a new maintenance work on a certain road is scheduled the operator has to insert a new event in the system along with
information as the starting date and the expected duration of works. As well as the operator has to update previous events whose expected finish date has not been respected and to delete the events out of date. Even though the framework helps the operators with the managements of the events, the operator is in charge of finding out all the events information and take in care their update and deletion.

![Operator Info](image)

**Fig. 3 - Use cases for a system operator**

### 3 The sensor networks

There are many sources of information the framework can use to evaluate the best current route, among these there are the geographic data about the roads coming from GIS systems, information about events inserted by the operators through the web configuration panel, the users preferences stored in the user profiles, the news about the traffic provided by the sensors installed along the roads, and the user location either taken from the user profile or from the application installed on the client device. If we consider these data as atomic entities none of them has enough information to evaluate the best route between a start point “A” and a destination “B”. Hence we need to correlate as much as possible these data the make the most accurate prediction of the best route. For example if we know the geographic data of a road and we know the status of the maintenance works on that road but we do not know anything about the traffic, we could give a wrong suggestion to the user. In fact we could wrongly tell the user to take a road full of traffic only because we see that road free from maintenance works. The need of coping with the heterogeneity of the data [1] format is satisfied by the Data Monitor developed in the PHDS (“A Middleware Infrastructure for Real-Time Processing of Heterogeneous Data Streams”) project. In fact thanks to the Data monitor we are able to correlate much different kinds of data, producing the most accurate prediction of road conditions. It is important noting that in the previous example the data on the traffic are useful to evaluate a more accurate route but they are not needful because the framework can still work and prudence an evaluation of the best route with a certain level of accuracy.

### 4 The real time data monitor

The data monitor is the core component of the engine, it relies on the main outcome of the research project (PHDS). The data monitor is in charge of collecting data from a sensor network and analyzing them in order to provide the engine with useful correlated information. One interesting aspect of the data monitor lies in the adopted technique for data processing. In such scenario, where it is required the ability to analyze huge amounts of data in real-time, the data monitor uses the stream processing paradigm [2] in order to infer all the information of interest from the data. In the presented framework the data monitor is used by the engine in order to find out eventual (not ordinary) troubles on the user prearranged route. In figure 4 is depicted one example for a use-case of the framework. In such a scenario there are a registered user who used to reach his destination through the registered route “A”, three alternative routes for the desired destination, some traffic on the route “A” and maintenance works on the route “B”. The sensor network data include the geospatial data of the routes (static data), the data on the traffic (hard real time data), the user location (hard real time data) and road maintenance works (soft real time data). When the user starts the application on his device, two alternative situations are possible. If the user’s device has a GPS receiver and the user wants to use his current location, a message with the user current location is sent to our server, this information is correlated with all the other data by the engine to calculate the best route among the available routes. Instead, if the user’s device has not a GPS receiver the engine uses the location inserted by the user as starting location when he registered the route. In both cases the server replies with a message with the best available route between the user’s location and the destination. At the moment the user starts the application, the engine runs the following pseudo algorithm:

When (new u is active)
startingLocation = u.location
destination = u.desiredDestination
for each (route between startingLocation and destination)
    score = correlateInfo()
return (route with best score)

QUERY correlateInfo:
INSERT route descriptor
SELECT traffic, work, route preference
FROM road sensors, user profile, road data
WHERE road=route

When the engine recognizes a new user, it process all the routes available between the user’s location (sent by the user or stored in the user’s profile) and the user’s...
destination (stored in the user’s profile), afterwards for each routes it runs the continuous query [3] “correlateInfo” to assign a score to each route and returns the route with the highest score. The score is given by a combination of three values:

- Route preference: in his profile the user classifies the alternative routes according to his preferences.
- Traffic: a parameter that represents the traffic intensity on the route.
- Work: a parameter that represents the status of possible maintenance works on the route.

For example in the scenario depicted in figure 4 the engine would suggest the user to take the route “C” because it is the best combination of user preferences, traffic intensity and maintenance works status.

5 A real implementation

A first prototype of the proposed framework has been implemented in Afragola [4], a small municipality in Napoli, a city in the South of Italy. In this pilot application the sensor networks included eight traffic sensors installed [5] on the main streets of Afragola, geographic data restricted to Afragola roads, provided by Google services, user profiles inserted by the users during the registration phase, information about the street maintenance works in Afragola inserted into the system by a team of ten operators. In the figure 5 is depicted the configuration panel used by the operators to set a new event about road maintenance works.
In the configuration panel an operator can register a new event selecting all the roads along which there will be the maintenance works. Figure 6 shows that part of the configuration panel used by the users to put in the system their favorite routes. When a user insert a start and a destination point the system proposes different routes until the user chooses his favorite one as the default route.

Finally in figure 7 is depicted the scenario where a user has to choose how to receive the information about the best route. The system shows a preview of the available formats for the message delivery and the user has to choose one or more of them.
6 Acknowledgements

Authors are grateful to Giuseppe Cozzolino and GianLuigi Spagnuolo for the numerous fruitful technical discussions, and for developing the bulk of the code.
This paper describes the results of a research activity conducted within the framework of the research project “A Middleware Infrastructure for Real-Time Processing of Heterogeneous Data Streams (PHDS)”. Authors wish to thank the technical staff of all partners participating in the PHDS project (namely: Synclab, Intesis Spa, HAL Software Srl, CINI, and ICAR): their contribution was key for properly steering the direction of the research activity, as well as for improving the quality of this paper.

References: