Challenges in Collecting and Using Mobile's Location-Based Data

Eli Packer, Tali Yatzkar-Haham, Alexander Kofman, Jonathan Bnayahu
IBM Research – Haifa
University of Haifa Campus
Mount Carmel
Haifa, Israel 31905
972-4-829-6211
{elip,tali,kofman, bnayahu}@il.ibm.com

ABSTRACT
In recent years we have been witness to rapidly growing adoption and spread of smart mobile devices, not only for personal use but also by enterprises. An important feature provided by majority of these devices is location awareness – ability to detect the device's location on Earth and use it in applications. For enterprises, there is tremendous value in ability to leverage location context in business logic. However, implementing location awareness in enterprise environments introduces new challenges, both technical and organizational. In this paper we present and discuss main technical challenges associated with building location aware enterprise software systems.

Categories and Subject Descriptors
C.1.3 [Processor Architectures]: Other Architecture Styles—Cellular architecture (e.g., mobile)

General Terms
Design, Standardization

Keywords
Space, time, spatiotemporal, location, mobile, services, visualization, decisions, rules

1. INTRODUCTION AND MOTIVATION
Suddenly your location matters — it is part of doing business. Nowadays, mobile devices (devices, in short) are commonly used in modern enterprise environments, and every end user is able to report information enriched with time and location data. This brings many new opportunities for empowering enterprise software systems: location, is an important piece of context enabling smarter, timely decisions, providing better value for customers and increasing the competitive ability of the business. The most interesting and valuable type of spatial data is information about moving objects (e.g. people, vehicles), which their location changes in time. Further in this document we’ll be using the term ‘spatiotemporal’ whenever talking about location context associated with moving objects or events.

As an example of an organization that relies heavily on spatiotemporal information, consider a car-sharing company that provides cars for rent for short periods of time. A sample high-level architecture for car sharing management software is presented in Figure 1. To use car sharing services, customers install the company’s mobile app on their smartphones. The app allows searching for a nearest available car that matches user’s preferences, ordering a car, reporting issues, and using other services provided by the company. In addition, the car sharing mobile app reports the location of the customer’s device, providing spatiotemporal context for all system transactions. With a monitoring system that is aware of the orders’ locations, the company can identify high demand for cars in a certain area and thus decide to transfer available cars there. Such a company would also handle car failures. When a failure is reported by the car system, a decision engine can automatically allocate the nearest available service representative to fix the problem. In addition, the monitoring system of the application can visualize the locations of the cars on a map to identify anomalies in a certain area and push alerts to drivers when they approach this area. Location data can also be used as input to a visual analytic tool that analyzes historical data and uses visualization to help the administrators define new rules, preventing undesirable situations ahead of time.

Figure 1. High-level architecture of the car sharing solution.
The above example shows how different parts of the described environment use location data to provide better value. Building such a system, in addition to commonly known design and development challenges, involves addressing location-specific aspects: location acquisition, managing spatiotemporal data, using spatiotemporal context in business logic, visualization of spatial data, detecting movement patterns, and many others. In absence of a common application platform providing support for the location-related aspects, the responsibility for dealing with complexity of these issues lies on the developer.

The rest of the paper is organized as follows: In the next section, we elaborate on the challenges associated with developing location aware enterprise applications. We continue by presenting related work in Section 3 and conclude in Section 4 with summary and discussion of future of the enterprise application middleware in regard to spatiotemporal awareness.

2. CHALLENGES

We explore the common technical questions developers have to address when building location aware enterprise software, such as the one illustrated in the previous section.

2.1 Resource Optimization

First, location should be detected and reported by the endpoints. In the above example, as in many modern systems, endpoints are smart mobile devices. There are many different types of mobile devices (different operating systems, sizes, computing power), but there is a constraint common to all of them: limited battery time. Listening to GPS events, open network sessions, intensive network traffic – all these consume battery resources, shortening device lifetime and leading to poor user experience. Therefore resource-intensive mechanisms should be carefully designed to achieve a sweet spot where important data is collected and timely processed at a reasonable battery consumption rate. Although, there is no ‘silver bullet’ solution, and different applications may require different design decisions, here are several points that are common enough to be considered for majority of applications:

- Reducing communication load. Communication is usually heavy and costly. Thus, finding ways to decrease communication volume is crucial. For instance, instead of sending one location at a time, a device could transmit multiple consecutive location samples. Sending multiple samples will not only reduce the number of communication iterations but will also save the constant information (user ID and more) that will be sent once in each call. Another technique is to wait until the application needs to send other data to the server and add the location information to that message. The size of a corresponding packet is a tradeoff between the communication cost and the availability of the data on the server side. The larger the latency that the server can tolerate, the larger this packet could be.

- Tradeoff between accuracy and efficiency. Reporting location updates would allow more accurate tracking on the server but would be less battery-efficient and would create more load on the network and on the backend components. Less frequent reporting would mean less accurate tracking as well as a delay in detection of important situations. Therefore decisions about when and what to report are essential to optimize the tradeoff between communication and resource initialization. These decisions should be dynamic and change as necessary in real-time.

- Location sampling heuristics. Although it is possible to collect location data at the maximal accuracy provided by the location sensor (e.g. GPS receiver) it turns that it’s not always necessary. For example, when a car moves at constant speed along a road, its location could be predicted pretty well for some time. Also, some mobile operating systems allow listening only to ‘significant’ location changes (definition of significant distance threshold is up to the developer) [11][12]. Developers can change the value of significant change threshold at runtime to improve accuracy at some places while providing just ‘good enough’ information in others.

- Deriving sample interpolation. A close challenge is to provide more information in between samples to allow deriving interpolated values. In the simplest form and by default, the interpolation should be linear in both time and space, but in other cases, more informative data, such as modeling the curve as a Bézier curve or a spline, could be useful in reducing the communication while providing valuable information on how to interpolate the samples.

2.2 Data Format

Another important aspect is the format and semantics of location data. While mostly application-dependent, we can still detect common data whose collection may be not trivial.

- Coordinate reference system. Most frequently, location is represented using geographic coordinates (longitude and latitude). However, in some cases other coordinate systems may be used in different parts of the system. For instance, some devices may use UTM coordinates, or use different representation formats for same coordinate system. For 2D map visualization UTM is the most common coordinate system, however different subtypes of UTM may be used for specific regions, such as continents, countries, or even different areas of same country. In other cases, it is possible that the system will use non-geographic coordinates, such as in cases in which the system is targeted to indoor environments. This point should be considered when architecting location aware systems. For instance, service APIs should provide simple and clear way to specify coordinates and coordinate system. Also, a well designed extensible system would allow adding support for new coordinate systems in future.

- Different location provider systems. Today's technology brings us various ways to collect location data; just to name a few: GPS, GLONASS, WIFI, RFID, and cellular networks. GPS and GLONASS provide geographic coordinates but don’t work indoors. WIFI and RFID allow capturing object’s location indoors but don’t provide geographic location directly. If it is required to seamlessly track location of objects both indoors and outdoors, the system must be able to combine location data coming from different sources (e.g. GPS and WIFI).

2.3 Data Processing Challenges

As location-aware data combine location with time, several important challenges arise. In traditional Geographic Information Systems, time was not a first-class-citizen. The addition of the time parameter as an integral part of the data and its analysis is far
from being immediate or trivial. On the contrary, it requires large modifications and adjustments, addressed in the following points.

- Several existing DBMS products provide spatial capabilities, either as inherent feature or as an extension package. A few examples are PostgreSQL, MySQL Spatial Extensions, DB2 Spatial Extender, Informix, MongoDB. These systems are capable of indexing large amounts of spatial data to allow spatial queries on that data in reasonable time. Unfortunately, most of those database systems do not support time as a special property that either constitutes an additive dimension (e.g., 3rd dimension for planar data) or is treated specially as a monotone piece of information. As opposed to space, consecutive time reports are monotone.

- There is a set of common problems that have to be resolved by the application developers and there is no commonly available platform that provides complete solution. The challenge here is to develop generic tools to establish suitable platforms. The following make up a small sampling of related challenges.

  - Processing trajectories. Numerous packages gather single location samples to trajectories and then analyze them and react accordingly. This functionality likely affects all of the solution components and requires corresponding adjustments.

  - Cluster data. It might be crucial for the solution to be able to hierarchically cluster reporting entities. For instance, a group of employees that report locations may work together towards a specific target, thus grouping their induced trajectories is necessary for coming up with important conclusions. Clearly, this clustering structure can be hierarchical, adding complexity to the data. As to the former point, this information might be essential for several components and thus induce interesting challenge to the platform design.

  - Discarding information. As the applications may collect enormous data, it might be crucial to devise heuristics for discarding old data. As devices report location in a timely manner, potentially samples become old and thus should be discarded or alternatively, sent to the archive. Clearly this process should be designed carefully and be consistent among the various components.

Another major challenge relates to the potential huge rate of location updates. A system that supports many entities that report their locations with high frequency and then computes complex online algorithms needs enough resources to meet the application requirements. Thus, both algorithmic challenges and strong hardware are required. The algorithmic challenges relate to providing asymptotic efficient solutions and possibly parallel computations.

### 2.4 Interface and Visualization

Many servers include administrators that control and configure the solution components. It is very important to provide the administrators with efficient interfaces so they can easily control the system and receive feedback. An important aspect in this regard is the visualization behind the interface, which is essential as space-time geographic information is involved. The visualization will most likely include maps on which relevant data will be displayed. More advanced scenarios will include the time dimension as well, thus visualizing the data in 3D, adding the corresponding major challenges that involve this upgrade. Today's technology (touch screen, voice recognition systems, etc.) brings new opportunities and challenges in this regard.

### 2.5 Uncertainty

Most geographic data (such as GPS data) are approximate to begin with. In many cases, evaluating the accuracy certainty level is possible, as a Gaussian distribution around a center point, for example. This information may be crucial for various analysis carried out by the analytic components. Whether given by the device or evaluated upon receiving the data at the server side, this information may be crucial and should be attached to the data along the process.

### 2.6 On-Device vs. On-Server Computations

It is often beneficial to delegate spatiotemporal computations to the mobile devices for the following reasons. First, this would reduce the load from the server side, thus increasing system’s scalability. Secondly, this improves application resiliency, allowing seamless operation when the network is unavailable. Third, it allows much shorter latency between event occurrences and their reactions. The following is a non-exhaustive list of related pointers on this issue.

- Whenever all the information required for a computation is available on the device – strive to performing the computation on the device
- Computation on historic data cannot be done on the device unless the history is kept on the device (which is mostly not true)
- Heavy computations that are expected to be too demanding for an average device to carry out should be delegated to the server (see resource consumption discussion). That said, a well designed system could devise an appropriate strategy dynamically, based on the device’s type, battery level, and possibly some other factors.
- When latency is critical – compute on the device, do not count on the link to server.

### 3. RELATED WORK

Prior related work relates to spatiotemporal support, both in the research community and in some commercial products, for the various components of a platform.

In the event processing field, Microsoft StreamInsight [2] provides support in the Microsoft SQL Server Spatial library, which is based on the Open Geospatial Consortium implementation specification [1]. Oracle event processing, which is also SQL-based, supports spatial operators, such as inclusion, in queries [5]. Moody et al. [7] defined a language for spatiotemporal event patterns, while a comprehensive discussion on spatiotemporal patterns was done by Barouni and Moulin [4]. In the mobile domain, [8] explains the notion of location-based services and describes the various aspects of their development, while [6] relates to location information as part of the broader area of context-aware computing. [9] is a nice example of a GIS analytic and visualization toolkit. We believe this work may assist similar efforts in their design and implementation phases.
4. CONCLUSIONS AND FUTURE WORK

Having available location data at hand provides enterprises’ applications with the ability to understand and respond to various spatiotemporal situations. However, constructing a system that receives a high volume of streaming spatiotemporal data that is further stored, analyzed, processed, visualized, and delivered is a challenging task. In this work, we proposed several useful pointers on how to construct a correspondingly complete and reliable system. In the future, we would like to continue this work by defining comprehensive and unified spatiotemporal services and algorithms that address the challenges addressed in this paper.

5. REFERENCES