TIME: An open platform for capturing, processing and delivering transport-related data

Jean Bacon*, Alastair R. Beresford*, David Evans*, David Ingram*, Niki Trigoni†, Alexandre Guitton† and Antonios Skordylis†

*Computer Laboratory, University of Cambridge
Cambridge, United Kingdom, CB3 0FD

†Computing Laboratory, University of Oxford
Oxford, United Kingdom, OX1 3QD

***Abstract*— Road congestion and traffic-related pollution have a large, negative social and economic impact, and we believe many of these problems can be reduced through investment in monitoring, distribution and processing of traffic information. This paper outlines how our on-going work on the TIME project (Transport Information Monitoring Environment) provides a solution, using traffic sensor systems and the design and development of an open and decentralised software framework. We also discuss how we address the privacy and security implications of the increased use of sensors and data processing.**

Keywords: event-based middleware, sensors, communications, privacy, access control

INTRODUCTION

The number of miles travelled by vehicles on the UK's roads has doubled in the last twenty-five years [1]. Road congestion is strongly correlated with increased road usage and now has a large negative impact on many economies throughout the world: Road congestion in the UK was estimated to cost the economy £12bn in 2004 [2]; similarly, the cost in the US was $63.1bn in 2003 [3].

We believe that investment in monitoring, distribution and processing of traffic information should result in a substantial and significant increase in transport efficiency. Such data gathering and processing should enable better strategic planning and encourage better use of public transport, both of which would help cut pollution and congestion.

In order to maximise the benefit derived from increased data gathering and processing we believe it is necessary to build an open platform allowing many companies and individuals to collaborate and share raw and processed data. Such a platform should enable a marketplace in which companies can securely share and sell gathered data in order to encourage investment in sensors, networking and processing facilities. It should also enable local residents and visitors to collaborate, calculating their own statistics and building their own applications.

The aim of our research in the TIME project (Transport Information Monitoring Environmant) is to: investigate, design and build suitable sensor and network technology; design and build reusable software components to distribute, process and store sensor data in real-time; and do both of these things with due regard to personal privacy and commercial interests. By this means we aim to make gathered data widely available to policy makers, application developers and citizens. The TIME project is focused on the city of Cambridge.

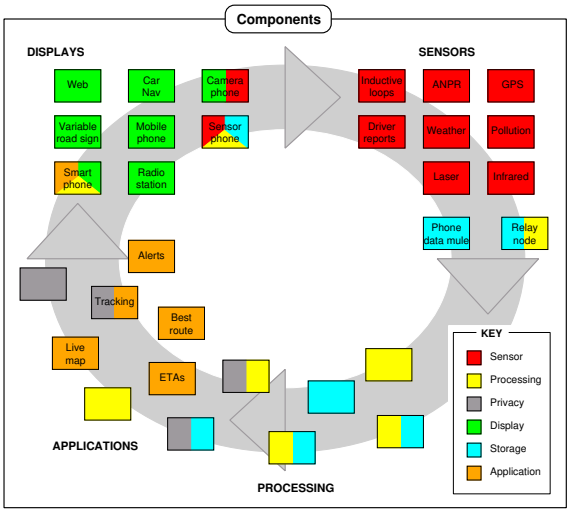

Fig. 1. Components and data life-cycle: sensor data flows through shared processing components to applications which place pertinent data on displays. Users viewing such data will then modify their behaviour, adapting future sensor readings accordingly.

We believe an open platform should enable a continuous life-cycle of data harvesting, processing and display, as shown in Figure 1. Such a system will enable more useful applications to be constructed by providing them with the data needed to make intelligent decisions. For example, offering real-time advice on the best route to the nearest airport often involves evaluating a rich set of alternatives: a direct motorway journey by private car or taxi; a journey first to the bus depot or railway station (by car, cycle or taxi) followed by a train or bus journey, and so on. In many medium-to-large cities there

is a multitude of rail, coach and other mass transit options available. Writing an application to make a continuously updated recommendation on the best route requires lots of data, including measurement of traffic flow on the road network, availability of parking, collation of the prevailing rail schedule, etc. Our aim is to support this type of application.

I. Sensor network

The sensor infrastructure is at the heart of the monitoring application; it all starts with the acquisition of sensor readings. In this section, we discuss the building blocks of a sensor network infrastructure for traffic monitoring. We explore various sensor technologies, propose an architecture, and report our initial experiences from managing real traffic data in the network.

Sensor network architecture: The existing traffic sensor deployment in Cambridge consists of 112 inductive loop sensors that generate flow and occupancy readings and propagate them to a central server along wired links.

Our goal is to extend this system in a non-trivial way, designing a sensor network architecture that is able to incorporate new sensing and communication technologies. We propose a hybrid architecture that consists of nodes with diverse communication, computation and sensing capabilities, connected through wireless or wired links. We distinguish the following three types of nodes:

Sensor nodes whether stationary or mobile, are location-aware and equipped with traffic sensor devices. Sensor nodes typically have limited communication, storage and processing capabilities. Certain sensor nodes (e.g. mobile phones) may be battery-powered and thus severely constrained.

Gateway nodes collect the readings from the sensor nodes. They have a fixed power source, plentiful bandwidth, storage and processing capabilities, and feed data to the middleware components for further processing and storage.

Relay nodes are deployed to ensure connectivity between sensor and gateway nodes. They are useful when sensor nodes are sparsely deployed, and cannot reach any of the gateway nodes through multihop wireless paths. Some relay nodes may be battery-powered, and thus severely energy-constrained.

Sensor node technologies: The currently deployed 112 inductive loop sensors generate traffic flow and road occupancy readings every 4 seconds and use wired links to propagate them to a central server. We plan to extend this existing infrastructure with emerging sensor technologies ranging from infrared sensors to GPS-equipped mobile phones.

We are using IRISYS thermal imaging camera systems provided by InfraRed Integrated Systems Ltd. to record vehicle volume and speed on urban roads in Cambridge. These units are relatively cheap, self-contained, are ruggedised for outdoor usage, and are compact. The unit determines vehicle volume and speed by running image processing algorithms on data from a low-resolution (16×16 pixel) thermal image sensor. Using a thermal sensor simplifies image processing by removing background image clutter, since vehicles and bicycles appear as bright objects on an otherwise uniform grey background.

Besides stationary sensor nodes, we are planning to use location-aware mobile devices, like GPS-equipped mobile phones, PDAs and car sensors. Our vision is to enable city residents to participate in traffic monitoring as sensors and as data mules. In terms of sensing, it is possible to analyze GPS trajectories of users travelling by car in order to measure car speed at different locations and infer from it other traffic measures, e.g. congestion, car flow and road occupancy. In terms of communication, we envision using Wi-Fi enabled mobile phones as data mules to relay traffic information in a delay-tolerant manner. The CarTel project [4] also uses mobile sensors deployed in cars to collect, process and store data. However, there are two major design factors that distinguish CarTel and our project. First, we assume an open architecture that allows the interconnection of several different users. Second, security and privacy issues are central in our project, and cannot be performed within the network.

In-network data management: In order to reduce communication, we investigate compressing traffic data within the network before propagating it to the gateway nodes. The framework enables the end user to specify accuracy requirements, which translate into the amount of data that is propagated; we believe most users are willing to tolerate small inaccuracies in the reported traffic data. We tested lossy compression techniques like the Fast Fourier Transform (FFT). We applied it to compress a single time-series of traffic data, and further studied the use of spatio-temporal correlations to reduce multiple time-series [5].

We observed that traffic data (e.g. car flow) exhibits strong temporal correlations. As a result, a sensor node is likely to find in its memory a previous day with very similar traffic, and be able to approximate today's readings as a linear function of the previous day's readings. This allows it to only propagate a few regression parameters to the gateway, rather than the entire time-series. In Figure 2, sensor node c detects that the time-series of the current day (Tuesday) is strongly correlated

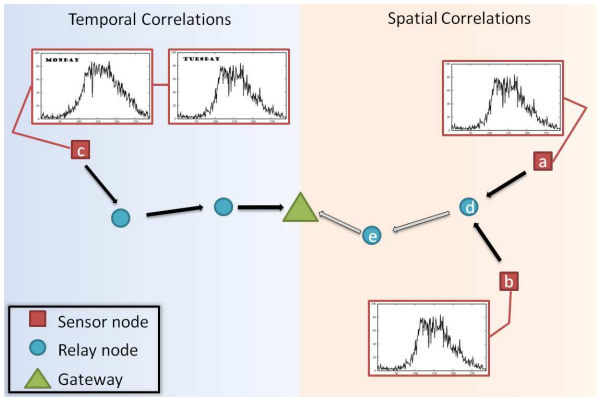

Fig. 2. Use of temporal and spatial correlations.

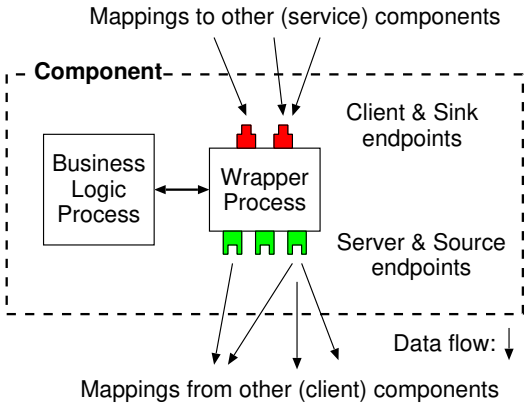

Fig. 3. Parts of a component

to Monday's time-series that the gateway already knows. It thus only forwards a set of regression parameters so that the gateway can derive the current time-series based on Monday's data, with a bounded maximum error as specified by the user.

Similar communication savings can be achieved by exploiting spatial correlations; Sensor nodes *a* and *b* in Figure 2 forward an FFT-compressed version of their time series to relay node *d*, the next hop on the way to the gateway. If a strong correlation between the two signals exists, node *d* expresses the time-series of node *a* as a linear function of the time series of node *b*; it subsequently forwards only the time-series data for node *b* and regression parameters for node *a* towards the gateway through the next hop *e*. A more detailed discussion is provided in [5].

II. Middleware

Once the data from sensors has reached a gateway node, it is passed on to the event-based middleware. At this stage it is worth examining the characteristics of the data, and the applications we wish to provide.

We observe that typical raw data streams from sensors are noisy, contain gaps, and commonly include erroneous readings due to sensor malfunction or miscalibration. For example, sometimes lanes in road junctions are moved but inductive loop sensors are not repositioned. Continuous service is interrupted when machines or links are reset, so in practice we consider breaks in the data to be the norm. These artefacts must be cleaned up before consumption by applications.

When considering applications that make use of sensor data, a common requirement is to combine multiple independent data sources. An application might wish to combine historical GPS trails and current induction loop data to derive estimated journey times around a city, or to fuse readings from two different but co-located sensors, for example. Furthermore our privacy requirements may introduce a layer of filtering between the user and some applications. We also need to accomodate occasional data format changes seamlessly.

We therefore need a middleware layer capable of distributing high bandwidth streams in real-time, and performing dynamic reconnection (to collect data from a changing set of sources, and distribute results to different classes of user or new applications, without halting the rest of the system). It will also be desirable to have subscriber (pull) interfaces so clients can request specific aggregate data, when an entire stream is unnecessary.

Established middleware such as object request brokers, web services and event systems are not very well suited to these tasks. We note however that our requirements are not extraordinary; they are shared by other sensor-driven systems not specific to road traffic, such as supply chain or environmental monitoring. To meet this emerging need we are constructing a flexible event-based middleware suitable for processing streams of sensor data. Our prototype is called SBUS (Stream Bus).

The system is decentralised and component-based. Data flow is peer-to-peer. Figure 1 gave some examples of typical components; the actual connections between them are not shown on this diagram. Each component may simultaneously act as a client of various other components and also as a server to others. Data from a sensor may be processed by a chain of components in turn before it reaches an application.

Figure 3 shows the parts inside an individual component. There are two separate processes; the business logic and the wrapper. The business logic is the application-specific part of the component. Typical tasks for the business logic include data reformatting or cleaning, calculating statistics, fusing data

sources, logging state changes, aggregation, pseudonymisation as well as sampling directly-connected sensors and presenting a user interface.

The wrapper manages a component's *endpoints*. All communication with components occurs via endpoints, whether data or control messages. Endpoints can be connected in a many-many fashion to the endpoints on other components. Server and source endpoints provide capabilities to other components, whereas client and sink endpoints represent the external resources a component needs in order to function. Each component can be "hotplugged" into the component graph by an appropriate mapping of endpoints.

The code for the wrapper process is the same for all components. The purpose of the wrapper is to combine the best properties of a centralised system with those of a peer-to-peer system. Centralised event broker architectures have the benefit of decoupling sources from sinks, whereas direct point-to-point links have lower latency and scale better by avoiding the central bottleneck. SBUS is decoupled *and* decentralised. For example, the wrapper can silently reconnect to required data source components if they fail and then reappear, avoiding the need for complex exception handing code inside the business logic in many cases. The wrapper also maintains lists of the other components connected to it, so it can distribute events to subscribed listeners.

Each endpoint comes with a list of roles required to connect to it. Normally when data from multiple streams are combined by one component, a stream it emits requires the union of the requisite roles. Role certificates are issued by components (not necessarily the same ones as the service providers) via a special endpoint in response to the presentation of credentials, or through administrator action. There are three kinds of access type: *private* (must not disclose data in any form), *redistributable* (may redistribute data providing the same roles are checked), and *republishable* (may republish data derived from this, and allow weaker sets of roles to view it, for example after anonymisation).

III. Privacy and Security

Applications constructed using this middleware will be context-aware, constructing from sensor data a model of the real-world environment that is used to serve their users. Effective context information, by necessity, includes information about people. This means that any transport application of non-trivial complexity will gather, collate, and distribute a tremendous quantity of personal information and could have a major impact on the privacy of individuals.

We believe that access control is well-suited to managing data having commercial value, but that collecting personal information and then protecting it does not adequately address privacy concerns. Even if permission to store data is obtained, it may be difficult to ensure correct protection. Understandable information must be provided by the system for users to give informed permission. Device configuration may be required, which places burden on the users. Furthermore, access control demands trust: submitting personal data to an application means trusting that access control constraints will be observed. As demonstrated by well-publicised leaks of personal information, such trust may be misplaced.

We are therefore exploring mechanisms by which we can eliminate personal information from the system, whilst still allowing transport monitoring applications to function. Such information includes still images and video, the capture and storage of which are common in modern monitoring systems. For example, automatic number plate recognition (ANPR) is a form of optical character recognition that extracts the digits of car number plates passing in front of a camera, producing a unique identifier for each vehicle that allows the tracking of movements through a network of cameras. While the use of such data for law enforcement may be justified, many transport monitoring applications have no need to remotely identify individual vehicles. In such cases we advocate the use of the equipment from IRISYS discussed in Section I. It provides little in the way of identifying information since the thermal image sensor cannot capture car registration data, yielding a privacy-sensitive means of counting vehicles, measuring vehicle speed, and estimating travel patterns.

Communication devices such as mobile phones provide another source of personally identifying information and communication between specific people may be identified, allowing reconstruction of individuals' social and business networks. So long as mobile phones are used for communication, we do not expect unique identifiers to disappear. Anonymous communications solutions developed for web browsing [6], messaging [7], applications using TCP [8], and even GSM mobile telephony [9] can be applied to mitigate the risk of determining which mobile device is communicating with a particular middleware component. We are exploring the capabilities of these systems when used to transmit sensor data that they provide. We expect that these messages will be small and will require low latency delivery, a combination that is not well addressed by the above systems.

The private information handled by transport monitoring ap-

plications is not limited to that collected by sensors or required by the communications infrastructure. The most natural route to modelling user state within an application's context may involve gathering information from the user that identifies him or her or that describes his or her condition. For example, an individual may wish to notify friends that it is time to meet at the pub. To control the distribution of this personal information, we are use context-specific pseudonyms [10]. Such a pseudonym blends an individual's identity with some assertion of his or her current context, such as "I am on the bus with number plate AE02 RYA" or "I am in the city centre". A pseudonym for user i within context n that provides for linkability by user j is expressed as $P_{i,j}^{(n)} = H_{K_{i,j}}(\mathrm{ID}_i, C_n)$ where $K_{i,j}$ is a key known only to users i and j and $H_K(\cdot)$ is a cryptographic hash function using key K. These pseudonyms may be placed in a database that can operate without access control because, without having the appropriate key, each pseudonym appears to be a random sequence of bits.

Data that are of commercial value include any sensor-derived information that may be sold or exchanged. For example, taxi companies frequently equip their fleets with GPS units and appropriate communications infrastructure to aid dispatch but must ensure that this information does not fall into competitors' hands. Nevertheless, such information could prove invaluable in determining traffic behaviour. We are developing an approach to protecting these data that translates business relationships between data providers and users into the access control functions supported by our middleware. This will yield what is essentially Digital Rights Management (DRM) functionality for transport data. The procedure will use the following steps: **(1)** Formal business relationships, describing how data may be processed and by whom, are captured and converted into a machine-readable representation. **(2)** These low-level representations of business relationships are mapped to individual data sources and assembled into licences that describe how each data source may be used. We intend to use the Rights Expression Language of the MPEG-21 Digital Rights Management framework for this task, as it provides a rich set of primitives for describing the operations that may be performed on digital data [11]. **(3)** The licences attached to the various data sources available will be used to assign roles to the components participating in the application and issue certificates as appropriate.

IV. Conclusion

We proposed an open platform for capturing, processing and delivering transport related data. Building such a platform presents many challenges. At the sensor network level, we described techniques for energy-efficient gathering of traffic sensor data. At the middleware level, we described a decentralised architecture which works in the face of individual component failure, and a strong interface binding between components to ensure applications get the data they were expecting. Data persistence is also needed to allow historical analysis and pattern matching to detect incidents and anomalies. At the user level, we discussed security and privacy issues. More specifically, we proposed methods to protect the financial interests of companies who paid for data collection as well as anonymisation techniques to protect the privacy of individuals.

Over the next few years we will continue to develop and integrate our technology with existing systems in Cambridge, England. Future work will also extend our system from monitoring to control. For example, can we use our data archive and real-time streams to improve traffic light scheduling? Can we effect a positive change—through the provision of real-time information—in the modes of transport taken by residents?

References

[1] DfT, "Traffic Statistics for Great Britain, chapter 7: Traffic – data tables," tech. rep., Department for Transport, October 2005.

[2] R. Devereux, J. Dawson, M. Dix, G. Hazel, D. Holmes, S. Glaister, S. Joseph, C. Macgowan, B. Nimick, M. Roberts, L. Searles, R. Turner, S. Gooding, S. Hickey, and W. Rickett, "Feasibility study of road pricing in the UK – report," tech. rep., Department for Transport, July 2004.

[3] D. Schrank and T. Lomax, "The 2005 urban mobility report," tech. rep., Texas Transportation Institute, May 2005.

[4] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden, "CarTel: a distributed mobile sensor computing system," in *SENSYS'06: Embedded Networked Sensor Systems*, November 2006.

[5] A. Guitton, A. Skordylis, and N. Trigoni, "Utilizing correlations to compress time-series in traffic monitoring sensor networks," in *WCNC'07: Wireless Communications and Networking Conference*, November 2007.

[6] M. K. Reiter and A. D. Rubin, "Crowds: anonymity for web transactions," *ACM Trans. Inf. Syst. Security*, vol. 1, no. 1, pp. 66–92, 1998.

[7] G. Danezis, R. Dingledine, and N. Mathewson, "Mixminion: Design of a type III anonymous remailer protocol," in *Proceedings of the 2003 IEEE Symposium on Security and Privacy*, May 2003.

[8] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in *Proceedings of 13th USENIX Security Symposium*, pp. 303–320, August 2004.

[9] D. Kesdogan, H. Federrath, A. Jerichow, and A. Pfitzmann, "Location management strategies increasing privacy in mobile communication systems," *Information systems security: facing the information society of the 21st century*, pp. 39–48, May 1996.

[10] D. Evans, A. R. Beresford, T. Burbridge, and A. Soppera, "Context-derived pseudonyms for protection of privacy in transport middleware and applications," in *Proceedings of the First International Workshop on Pervasive Transportation Systems*, March 2007.

[11] International Standards Organisation, "Information technology—multimedia framework (MPEG-21)—part 5: Rights expression language." ISO/IEC 21000–5:2004.