Employing HUMS to Automate Dull Administration Tasks in Logistics and Fleet Management

Guy Edward Gallasch, Nicholas Brealey and Leong Yen

Land Division, Defence Science and Technology Group,
PO Box 1500, Edinburgh, South Australia, 5111, Australia

Abstract

Automation, autonomy and autonomous systems (AAA) promise enhanced and improved logistics operations and fleet management, whilst reducing related administrative burden. In order to begin exploration of AAA, and in particular alleviating dull administration tasks, we embarked on a six-month pilot study to implement a system to automate the collection of land vehicle health and usage data. Commercial solutions for achieving this are abundant; hence the objective of this study was not to reproduce that which already exists. Rather, the study was designed with two objectives. First and foremost, this study was designed to be an internal skills and capability-building exercise, to gain experience in undertaking short, focussed prototyping activities with both hardware and software components. The second objective was to produce a physical test-bed for future Health and Usage Monitoring System (HUMS) experimentation and prototyping. Accordingly, this paper reports in brief on the system developed, but more so on the experience gained and lessons learnt in conducting such short, highly focussed prototyping activities. We hope that sharing our experience will provide valuable insight to HUMS researchers considering undertaking similar studies.

Keywords: Vehicle Health and Usage Monitoring Systems, Automation of Administration, OBD-II, CAN Bus, Prototyping, Proof of Concept development

Introduction

Dull administrative tasks are not generally enjoyable and are susceptible to neglect in the face of (perceived) higher priority tasks. This is particularly true when the benefits of undertaking the task are not immediately evident. As most administrative tasks boil down to data collection, the potential for Health and Usage Monitoring Systems (HUMS) to address poor data collection practices and outcomes is significant. However, great care must be taken with the introduction of any sort of automated data collection capability. It must be reliable, robust, effective at collecting the data required, and must do so in a timely manner. Above all, there should be no (or minimal) additional impost and an overall reduction in administrative burden.

To this end, the Land Logistics group within Land Division of DST Group embarked on a six-month pilot study into the automation of vehicle health and usage data collection. This pilot study represented our initial foray into the exploration of automation, autonomy and autonomous systems (AAA) in Army logistics. Our pilot study had both a primary and secondary objective:

- Primary: to learn and develop skills in prototyping and developing proofs-of-concept in both hardware and software, and developing 'hands-on' experience in undertaking short, time-constrained, and highly focussed activities of this nature.

- Secondary: to produce a HUMS physical test-bed for vehicle health and usage data collection that is suitable for rapid prototyping activities (e.g. readily customisable and extensible) in the exploration of AAA ideas and concepts.

To provide focus to our pilot study, automating the data collection for the "AD049 Vehicle Authorisation and Task Form" was proposed, having been previously identified as being of interest to Army. AD049 is essentially a vehicle usage log, and is representative of the type of 'dull' administrative task to which existing vehicle HUMS can be applied.

This paper briefly documents the design and implementation of our system for automating AD049 data collection. As per our primary objective, it was not our intention to produce a polished end product for deployment across fleets, as clearly there are commercial solutions that could provide this capability (although unsuitable for a test-bed). As such, the discussion in this paper focuses primarily on the learning outcomes from having undertaken this six-month study, rather than on the specific hardware and software solution developed. More details on the hardware and software aspects can be found in [1].

AD049 – An Example of a 'Dull' Administration Task

In brief, data collected by the AD049 form falls into three broad categories: 1) vehicle authorization (identity of the custodian, date and duration of allocation, and identity of the authorising officer); 2) Petrol, Oils and Lubricants (POL) usage (date and time, POL type, volume, supplier, and docket number); and 3) vehicle task details (date, start and finish times for each trip, start and finish odometer reading for each trip, a brief description of the purpose of each trip, whether the trip was business or private, and identity and signature of the driver). AD049 data is used for reconciling fuel purchased using Defence fuel cards, identifying drivers for reconciliation of infringement notices, and compiling fuel consumption data for unit transport managers, and Army's Joint Fuels and Lubricants Agency.

Clearly, some aspects of AD049 are difficult to automate, such as determining the purpose of a trip. A relatively easy task in comparison is collection of data on refuelling and vehicle tasking. We have targeted four areas:

1. Time and duration of each trip, via Global Positioning System (GPS) time signals (if available) or an embedded real-time clock within the HUMS data collection device.
2. Trip distance, which can be taken directly from the vehicle's On-Board Diagnostics (OBD) system through the exposed OBD-II[1] port.
3. Fuel consumed[2], based on well-known real-time fuel consumption calculations (e.g. see [2]), again supported by data from OBD-II queries.
4. The use of Radio Frequency Identification (RFID) tags as a proxy for a more sophisticated driver identification mechanism.

In addition, although not mandated explicitly by AD049, we considered including a tyre health monitoring capability, for two reasons: vehicle occupant safety, and to demonstrate the collection of additional HUMS data with negligible additional ongoing cost. A full discussion of alternatives and implementation options for AD049 data collection is given in [1].

[1] OBD version 2, the current industry-wide standard adopted for vehicle on-board diagnostics systems.
[2] Not fuel purchased, as 1) automatically recording the volume purchased would require either fuel bowser or vehicle augmentation; and 2) it is specific to the vehicle itself and hence excludes fuel purchased for other uses.

Implementation

Overview and System Architecture

For pragmatic reasons, we targeted the 'white fleet' Land Division divisional vehicle. Our proof-of-concept required mechanisms to collect the targeted HUMS data, record and analyse it, and communicate it to an off-board system for storage. The diagram in Fig. 1 provides a high-level view of the final system architecture. There are four main components: 1) an OBD-II data collector (ODC); 2) tyre Temperature and Pressure Monitoring System (TPMS) sensors; 3) Vehicle Data Coordinator (VDC); and 4) Central Data Repository (CDR). Each is described in brief below.

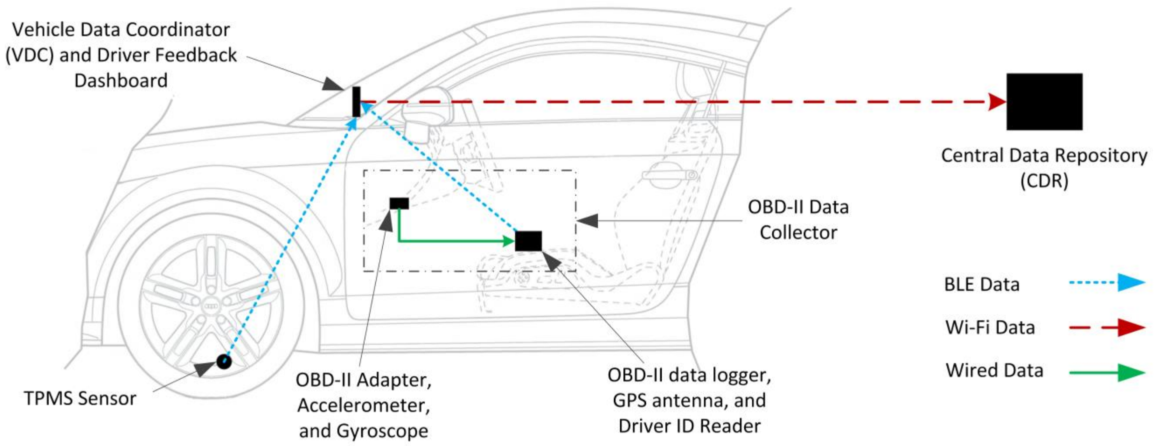

Fig. 1: Our vehicle HUMS system architecture (background image via [3])

Collecting On Board Diagnostic Data: the OBD-II Data Collector

For the collection of vehicle OBD data, we selected a Freematics OBD-II data logger [4] because both the hardware and software are open-source, hence (in theory at least) is rapidly customisable and extensible. We selected the Telematics Kit #2 [5], pictured in Fig. 2, which comprises an Arduino-based OBD-II data logger, OBD-II adapter (plugs in to the vehicle's OBD-II port), and a GPS receiver on a fly-lead. The Freematics solution also provided an accelerometer, gyroscope, Bluetooth Low Energy (BLE) communications, GPS, SD memory card logger and graphical display for displaying vehicle data in real-time.

Fig. 2: Assembled Freematics OBD-II Telematics Kit #2, including external GPS receiver

Due to a possible hardware design fault in the logger, and as an additional and significant learning exercise in its own right, we decided early on to build our own hardware solution from generic Arduino boards and expansion boards (so-called 'shields') and to develop the required software. The hardware solution was further augmented with a passive RFID reader, real-time clock, small Lithium Polymer battery and charging circuitry, and an enhanced power regulator. Although we still used the Freematics firmware as a base, extensive modifications were required for it to work with the hardware solution that we call 'Frankenstein', pictured in Fig. 3, with further modifications to perform the fuel consumption calculations within the OBD data logger itself.

As per the architecture in Fig. 1, we explored the use and integration of a TPMS. We were unable to find an open-source TPMS. Although we found that it was possible to integrate a commercial BLE-based TPMS into our custom hardware, to provide a data feed for the purposes of proof-of-concept a BLE "proxy" TPMS was proposed, but was not implemented.

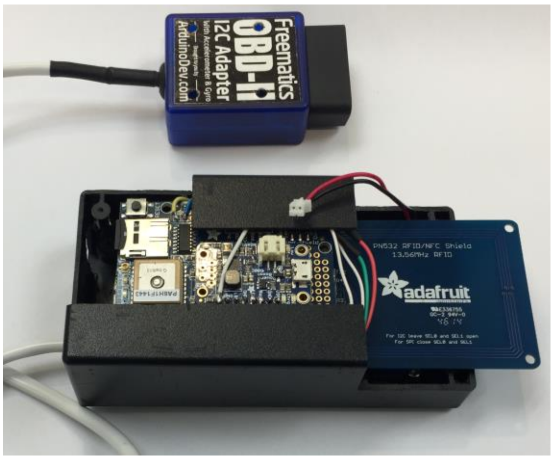

Fig. 3: Our custom-built OBD-II data logger, pictured with Freematics OBD-II Adapter

Tying it Together: the Vehicle Data Coordinator

As per Fig. 1, we have implemented a VDC that integrates data from multiple sources (in our case, the OBD-II data logger and the four tyre sensors) and acts as a gateway to the central data repository. A smart device (initially an iPhone, but later a Samsung Galaxy Tab) provided a suitable pre-built solution with significant processing and data storage capabilities, communications (Bluetooth, wireless Ethernet, and often cellular), and a high resolution screen. The software 'app' we developed coordinates communication with the OBD-II logger via BLE, and communication with the central data repository (see below) via WiFi, providing real-time streaming of OBD and tyre data when there is connectivity, and buffering this data when there is not. Further, the app also used engine RPM to detect the start and end of trips and produces a textual record of the trip (forwarded to the central data repository) that incorporates the AD049 information of interest identified above.

Some level of high-level graphical feedback was also able to be built into the app within the six-month time window, including graphical indications of vehicle tilt and roll, and tyre condition. However, this was a lower priority than implementing the underlying connectivity for communication of data, and so the visualisation remains rudimentary at best. More details of the proposed driver feedback dashboard and visualisations can be found in [1].

Exporting the Collected Data: the Central Data Repository

At the time of writing, our CDR was a Microsoft Surface Pro 3, running a Java server that accepts incoming connections from the VDC over a wireless Ethernet connection. Although a Surface Pro may appear to be an odd choice to act as a proxy for a database and analytics system, we are keen to demonstrate that the CDR is also an information provider and not just a static data repository. To this end, we wanted a device that could collect and store data as well as provide a platform to showcase various user interactions. In particular, we expect that information from a CDR would be used by a workshop manager to obtain vehicle-specific information, and a tablet is the ideal interface in such an environment. Further, the Surface Pro is able to function as a typical office computer and can be used to as an analogue to demonstrate how a fleet manager would access information from the CDR.

Most of the effort within the six months was devoted to developing the other components of the system, hence there has been little development of the user interface or analytic capabilities of the CDR. Further, consolidation of HUMS data takes place via text files and not within a relational database as originally envisaged, and the development of algorithms to analyse the data and present relevant information suitable to a range of users (such as fuel consumed while idling) was not able to be undertaken within the six-month time window.

Discussion

Despite the software aspects not progressing as far as we desired, by the end of the six-months we had demonstrated a solution capable of automatically collecting HUMS data from vehicles, performing analytics (e.g. real-time fuel consumption calculations), providing real-time visual feedback to vehicle operators, and consolidating all HUMS data into a central repository. Further, a summary of vehicle data pertinent to the automation of AD049 was able to be produced for each trip.

More importantly, we fulfilled our primary objective of experiential learning and skills development around conducting short, focussed prototyping studies. The more significant lessons learnt and insights gained are given below. Some of these were identified during the study, while others were identified upon post-study reflection:

- Be clear as to why you are doing this in the first place. For us, it was first and foremost as a learning experience in conducting studies of this nature, so in a sense the specific application tackled wasn’t crucial, but we still wanted it to be relevant.
- Problem and scope definition: be clear up front on the problem you are tackling, and what is in and out of scope. If this is not possible (e.g. due to the nature of the problem) have processes in place to iteratively refine these as the study progresses.
- Manage scope creep. Stay focussed, and don’t be tempted to expand the feature set of the end product unless necessary. The prototype doesn’t need to do everything, or be everything. If initial prototyping is successful, features can be added in later iterations.
- Communication: Continually record and disseminate to the team the progress of the project, the successes, difficulties, failures and lessons learnt as they emerge.
- Have realistic expectations of what is achievable in a short period. This includes being realistic about the amount of effort that can be dedicated to the task over its intended duration. We suffered to some extent with having other tasks of competing priority.
- Be timely, otherwise the outcomes of your work risk losing impact. Focus initially on quick wins and a quick turnaround on problems of interest to clients.

- Be cognisant of institutional barriers, so they can be taken into account when planning. These may include administrative overheads (e.g. procurement, approvals) and organisational/cultural barriers (e.g. support for activities that haven't traditionally been undertaken by specific work areas).
- Be aware that components designed with standardisation in mind, such as Arduino microcontroller boards and shields, may not all conform to the same 'standard.' We found our Arduino shields were incompatible with each other and with certain Arduino boards without physical modification due to pin allocation conflicts.
- Buy good quality components where reasonable to do so – don't agonise over spending $5.00 instead of $3.00, as that extra $2.00 may well save hundreds of dollars in lost labour costs sunk into debugging and/or repair and remediation work.
- Don't be afraid to build a prototype that looks like a prototype. Duct tape and cable ties are okay. Prototypes and proofs-of-concept need to be functional not necessarily pretty.
- It's okay to fail, and if you do, it is essential to fail fast, learn lessons, and move on.

Conclusions

Within our six-month pilot study we successfully achieved the outcomes and objectives we set for ourselves: 1) undertaking of an internal learning and skills development exercise in running short, focussed prototyping studies; and 2) producing a first-cut proof-of-concept physical test-bed for collection and analysis of vehicle HUMS data that is customizable, extensible, and supports the exploration of AAA concepts and ideas. Further, we have demonstrated that HUMS can be applied effectively to reduce the burden of dull administration.

Future work specifically on HUMS data collection includes a desire to improve and expand the capabilities of the hardware test-bed, both for collecting vehicle data and health data more generally (e.g. soldier health data). We are actively working to build capability in data analytic and machine learning techniques, with HUMS as one possible target. More generally, we intend to carry out more short, focussed studies such as this in support of our exploration of the utility and application of automation, autonomy and autonomous systems for logistics and fleet management. Ultimately, we are working toward creating a vision of 'digitised logistics' for dispersed, disaggregated operations, within which effective HUMS will play a crucial role.

References

1. G. E. Gallasch, N. Brealey and L. Yen, "Exploring Automation of Dull Administration in Logistics and Fleet Management: A pilot study on Prototyping and Proof-of-Concept Development", *DST Group Technical Report*, awaiting approval, 2017, 113 pages
2. Lightner, B. D. (2005). "AVR-Based Fuel Consumption Gauge." *Circuit Cellular* 183, 59-67
3. Cartype (2017), "Car Line Art." Retrieved 31 July 2015, from http://cartype.com/pages/4507/car_line_art
4. Freematics Hardware Store. (2015). "OBD-II Telematics Kit #2 (Arduino Compatible)." Retrieved 29 July 2015, from http://freematics.com/store/index.php?route=product/product&path=24&product_id=57
5. Freematics.com. (2015). "Freematics - Vehicle telematics made easy with open-source hardware." Retrieved 29 July 2015, from http://freematics.com