

Policy-enabled Internet of Things Deployable Platforms for Vaccine Cold Chains

Ioannis Daskalopoulos
University College London
i.daskalopoulos@ucl.ac.uk

George Roussos
Birkbeck, University of London
gr@dcs.bbk.ac.uk

Mohamed Ahmed
NEC Labs Europe
mohamed.ahmed@neclab.eu

Tony Delamothe
British Medical Journal
tdelamothe@bmj.com

Leo Brown
Netfuse Telecom
leo@netfuse.org

Stephen Hailes
University College London
s.hailes@ucl.ac.uk

Jagun Kwon
University College London
jagun.kwon@ucl.ac.uk

ABSTRACT

Cost effectiveness is critical to the success of immunisation programmes in developing countries. Being able to continuously, flexibly and accurately monitor the vaccine supply chain can help reduce the cost of operating it, and savings can be applied to deliver further resources to those that need them. In this paper, we present a Cloud-based continuous monitoring application for supervising the operation of the vaccine supply chain. The application utilises Internet of Things (IoT) location-aware platforms for data harvesting.

Our contribution to this area is the design and development of a prototype continuous monitoring device capable of adapting its behaviour according to both the contexts it encounters as it passes along the vaccine supply chain and the constraints that it is given. Our prototype is sufficiently flexible to operate at different points in the supply chain, and can be used to track shipments and provide real-time remote monitoring.

Keywords

Vaccine cold-chain, context adaptive behaviour, modes of operation, Internet of Things, temperature monitoring.

1. INTRODUCTION

It is reported that every 20 seconds a child loses their life due to a vaccine-preventable disease [1]. Since its foundation in 2000, the GAVI [2] alliance has helped to deliver vaccination to more than 370 million children in developing countries. In addition, with others, GAVI has worked to improve the vaccine supply chain and to minimise inefficiencies that can lead to wastage of vaccines.

Vaccines are produced in the developed world and shipped to many small clinics in remote places in the developing world. In doing so, they must travel by air, by sea, and over land, through

countries with different political systems, laws, and bureaucracies. On their way they will reside in different sizes and types of storage facility, with different management practices, for different amounts of time, in places with uncertain power supply and chillers of unknown reliability, and must arrive in clinic and be used within the use-by date, never having been outside a given temperature range. There is limited flow of information from the clinic up the vaccine supply chain and, consequently, it is a significant challenge to determine the efficiency of the system, either in whole or in part.

Over the past 10 years, the resistance of vaccines to temperature variations has been improved [3] and a wide variety of techniques and sensors (chemical and electronic) to monitor them have been developed and deployed [4]. However, at present, in many developing countries, inventory tracking and performance monitoring is still paper-based [5, 6]. Other recent actions to improve the vaccine supply chain have been directed at monitoring the performance of equipment such as refrigerators and cold rooms, as well as characterising the performance of the complete supply chain [7, 3]. However, the reality remains that there is limited deployment of sensors and automatic reporting tools that could help remote monitoring, especially with older assets. As a result, managing the vaccine supply chain assets is currently complex, expensive and error prone. Further, databases and analysis tools to aggregate and better analyse the monitoring data that does exist, for example to support fault localisation and prediction, are not widely deployed. This makes tasks such as planning for repairs, or strategically placing assets, difficult and manpower intensive.

The problem of vaccine supply chain monitoring in real time is further complicated by the dynamicity of the application environment and the tasks involved. For example, shipments cross regulatory and organisational boundaries and, as a consequence, become subject to different laws, some of which may prohibit the gathering or transmission of sensor information whilst in that jurisdiction (e.g. in some countries or in flight). At points, the shipments may be held for significant periods in one location, with a strong requirement for sensors to minimise power consumption, or be in transit, for which much more aggressive monitoring is required.

To have a real chance of being widely deployed, solutions to support the real time continuous monitoring of the vaccine supply chain must: i) enable users to define legal or operational constraints; ii) be flexible, capable of adapting dynamically to their environment; iii) be robust to failure; and, finally, iv) ensure that any adaptation does not itself prevent the system from working as

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MobiQuitous December 2-5, 2014 London, UK

Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...\$15.00.

expected. In this paper, we propose an adaptive continuous sensing application aimed at functioning across the vaccine supply chain. The application has been developed using an event-driven approach and is capable of dynamically adapting to its operating environment. Further, application modes can be configured easily to specify local policy (legislative/organisational) constraints, such as: “do not use the radio in location X”, or “be robust in localisation by using hardware (GPS/GPRS radio) redundantly”. Finally, policies can be validated to ensure that the application will behave as expected.

2. MOTIVATION

Most vaccines are temperature sensitive: they may not be held for significant periods above a given bound, and they must not be frozen. In order to reduce waste and optimise the supply chain, the World Health Organisation (WHO) temperature monitoring protocol [8] emphasises the need to maintain an accurate picture of operations through the whole vaccine delivery supply chain. Kartoglu and Milstien [3] highlight that the WHO guidelines for continuous monitoring of infrastructure are there to support the assessment and management of the risks involved in operating a cost-efficient health care system.

Unfortunately, accidental freezing occurs in all parts of the cold chain [9, 3]: 14-35% of refrigerators or transport shipments expose vaccines to freezing temperatures ($\leq 0^{\circ}\text{C}$) [9]. Further, studies of parts of vaccine supply chain in both developing and developed countries show that 75-100% of the vaccine shipments were exposed to freezing temperatures at least once within the distribution process [9].

Studies in regions of western China [10], Thailand [11] and Papua New Guinea [12], show that the typical shipment includes numerous changes in location, mode of transport and environmental conditions. Figure 1 (from [10]) illustrates one such example, and gives the temperature log of a vaccine the delivery to a remote region in western China using an outside cold chain. Ren et.al. [10] show that such changes in the operating environment, such as location or transportation mode, are common and contribute to observed temperature fluctuations.

Temperature fluctuations can also result from problems in supply chain management and practice, such as the improper utilisation of infrastructure (i.e. refrigerators) and the level of training received by practitioners [5]. Naik et. al. [13] give examples of how local practices may undermine quality assurance standards, such as not having independent voltage stabilisers attached to refrigeration equipment or not removing ice build up. Such practices have an impact on the performance of the vaccine supply chain as a whole.

Finally, to support continuous monitoring of the vaccine supply chain, the WHO provides recommendations for where and when to deploy devices, as listed in Table 1; sample devices are given in [4]. In current best practice, performing continuous monitoring involves calibrating and manually integrating the information from a variety of sensors from across the supply chain, many of which either do not support remote logging or are single use. This can be a man-power hungry and potentially error-prone process [3].

2.1 Technological Constraints

The vaccine supply chain incorporates numerous different environments and conditions, such as air/land transport, varying forms of storage, etc. This makes continuous monitoring applications, capable of operating across most of the supply chain, a challenge to design and deploy.

The WHO already provides guidelines for electronic monitoring devices in the vaccine supply chain [14]. In addition to these, be-

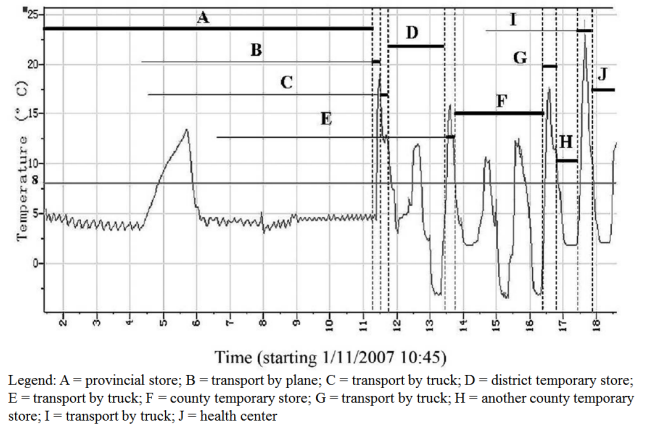


Figure 1: Example of temperature record provided by the data logger for vaccine delivery during one route, outside the cold chain, in remote regions of western China, from [10].

cause continuous monitoring devices are required regularly to cross international/agency boundaries and to change operating environments, we can expect that they regularly encounter limited and sporadically available resources (e.g. limited network availability, loss of GPS, etc.). A failure to recognise this may result in high power consumption as devices repeatedly attempt (and repeatedly fail) to connect. Consequently, continuous monitoring applications must be robust to the sporadic availability of resources and must provide the means to adjust their behaviour flexibly and dynamically. This might include, for example, utilising partially redundant information from different hardware to allow decisions to be made.

The current wide variety of deployed devices in the vaccine supply chain [4] can be attributed to a desire to maximise suitability for purpose with respect to the cost of deployment. For example, cheap feature-poor monitoring devices (such as electronic display tags [4]) may seem preferable from a cost perspective. However, the programmability and re-usability that comparatively more capable platforms offer can be attractive [6]. Furthermore, deploying a single device that can operate over much of the supply chain and automatically report its measurements to a centralised location, can reduce costs elsewhere. It helps simplify deployment by standardising the monitoring platform and can reduce the occurrence of error that results from mishandling equipment and misreporting results.

2.2 Organisational and regulatory constraints

All the agencies that handle vaccines in the supply chain are required to meet global regulatory requirements [3]. The countries involved in the supply chain issue laws and guidelines for operation, e.g. temperature control requirements on storage and transportation, operation of equipment, training, requirements on contractors, etc. [7, 15]. Such local regulations can have a big impact on defining compliant applications and platforms. For example, the use of wireless technologies may be limited while in flight or at sensitive locations. Given the potential sensitivity of the type of data that may be collected, some countries have regulations that determine what can be monitored or shared and when. Further, private contractors in the supply chain may impose additional constraints in order to comply with regulation or implement their operation (c.f. [16, 14]). For example, carriers may be unwilling to accept active monitoring devices on freight and/or unwilling to share data that

Description	Int. transport	Pri. vaccine store	Transport	Int. vaccine store	Transport	Service level
Electronic shipping indicators	✓		✓			
Vaccine cold chain monitor	✓		✓			
Vaccine vial monitor	✓	✓	✓	✓	✓	✓
Irreversible freeze indicator	✓	✓	✓	✓	✓	✓
Programmable electronic temperature and event logger systems with integral alarm and autodialer options		✓		✓		✓
Integrated electronic maximum/minimum thermometer, with factory-programmed alarms, for vaccine refrigerators and freezers		✓		✓		✓
Wall-mounted pen recording thermometer		✓		✓		✓
User-programmable temperature data loggers	✓	✓	✓	✓	✓	✓
30-day electronic refrigerator temperature logger		✓		✓		✓

Table 1: WHO-recommended temperature monitoring devices for storage and transportation of vaccines from [3]

may expose their logistics.

Amongst other factors, it is reported that differences in supply chains between countries (for example the occurrence of freezing temperatures is 16.7% in developed countries, and 35.3% in developing countries [9]) reflect the ways in which governments manage them [17]. Contrary to best practice, guidelines such as preparing ice-packs for cold box transportation are not routinely followed [9]. Studies on the performance of the vaccine supply chain (especially in developing regions) [13, 5, 9, 11] frequently cite inadequate training of practitioners as a major source of supply chain inefficiency.

With respect to the problems discussed so far, we identify the following challenges to realising widely deployable continuous sensing applications and devices:

Problem 1. The development of an easily configurable and extensible sensing platform

Problem 2. The development of flexible software to support the definition and execution of applications that adapt to the context/environment

Problem 3. The development of support for monitoring older/legacy devices in environments with limited resources

Problem 4. The integration of data from across the supply chain to help quickly identify and localise anomalies and validate that regulations are met

3. ADAPTIVE CONTINUOUS SENSING

3.1 A typical vaccine path

A monitoring device that accompanies a vaccine shipment enters the cold chain at the vaccine manufacturer’s premises [18]. Inside buildings, the GPS signal is usually absent but a GSM signal is usually available. The shipment is loaded into refrigerated vehicles where again the GPS signal is probably absent due to surrounding insulation but the GSM signal is usually available while in transit. With respect to monitoring, storage and transmission of data at this point, the consent of the manufacturer and the carrier is required in advance. Depending on the location of the manufacturing site and the legislation applicable, data transmission outside the country may not be allowed. Sampling and transmission rates can be frequent as the transmission costs are expected to be low as manufacturing usually takes place in developed countries [18].

Shipments are taken to airports where vaccines are loaded onto aircraft. By the time a device is located inside an aircraft it is required to enter flight mode (GSM and GPS off). Transmission of data cannot therefore take place from this point until the destination airport is reached; however, recording of temperature and other data

could potentially take place. Upon arrival at the destination, the vaccines are transported initially to a National Store where they are placed in refrigerators or in a cold room, if available [19]. When the shipment leave the aircraft, GPS and GSM usage can be resumed but the storage and/or transmission of the temperature, location and other data is now restricted by what is allowed within the destination country (legislation, cost of transmission etc.) giving rise, in turn, to a frequency of sampling and transmission issue [17, 20]. It is therefore necessary prior to any device activity to conform to what applicable legislation, context and needs dictate.

National Stores maintain a four month stock of vaccines and, typically, every 3 months the shipment is broken down and transported to Regional/Province Stores [18, 19]. As a result, a monitoring device may remain inside a Cold Room or a refrigerator, inside a building, lacking GPS signal for most of this time. In that state the device could use GSM functionality to retrieve cell tower location data and to transmit, if possible and permitted, the data harvested. Further down the Cold Chain, the vaccines are placed in Vaccine Carriers and Cold Boxes to be transported by any means available (car, motorcycle, public transport, bicycles etc.) from National to Regional Stores which further break down the shipment typically every month [19]. Regional Stores, in turn, transport to District Stores by using Cold Boxes. The ultimate destinations are the Health Centres to which the District Stores distribute their received shipment immediately upon reception. The point of administration in some cases can be outside the Health Centre, where a Vaccine carrier using a Cool Box transports the vaccines required for a particular outreach session [18].

3.2 Our approach

With respect to the problems listed in the previous section and the above scenario, we propose to develop an adaptive, continuous sensing application that can easily be deployed across the vaccine supply chain. It can be configured to conduct long-term measurements at a single location, such as in storage, or to measure across the vaccine supply-chain by accompanying shipments. To satisfy these requirements, the application must be flexible and adaptive. In other words, it must be able to change its mode of operation in response to some stimulus, such as: (i) dynamically switching between power saving or aggressive sensing mode, or, (ii) enabling/disabling various radio modules (GPS/GSM and etc) in response to its location.

The application platform is illustrated in Figure 2, and consists of the u-blox C027 platform, plus various sensors, including temperature, GPS and GSM radio, plus local storage. The basic function of the continuous sensing application is to periodically measure the current temperature and location of the shipment and to transfer the data collected to specified storage in the cloud, in the face of sporadic GPS/GSM coverage.

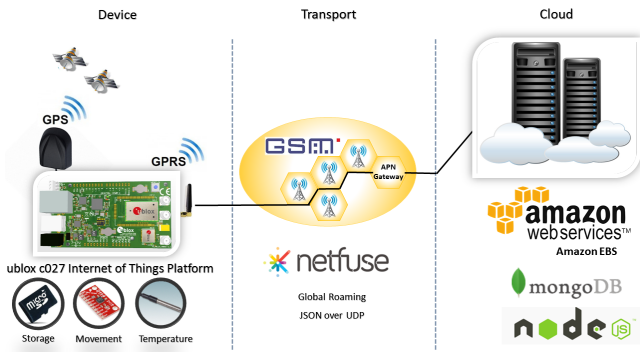


Figure 2: System Architecture. Deployable device, Data Transport, Cloud

The application is configured by first specifying the desired modes of operation, such as how the application should respond when the shipment reaches storage, as well as how to identify the event. Modes of operation enable the application to be versatile by specialising for different eventualities, such as disabling the GSM radio when in flight but not for other modes of transport.

Once the data is gathered, it is uploaded to local/remote storage. To do so, readings are serialised in JSON format, and packaged for transmission. When and how transmissions are scheduled can be determined flexibly. For example, the decision to transmit readings may be delayed in order to aggregate data so as to maximise available bandwidth, or in reaction to loss of network coverage. The decision to transmit via an SMS message or over a UDP connection can be made in response to the type of available coverage. Further, if there is cooperation with an operator, such as Netfuse [21], the network can be configured to perform in-network processing on the application traffic, for example, to enhance localisation by adding extra meta-data and provide specialised handling of the data including routing, encryption, anonymisation or compression, before storage.

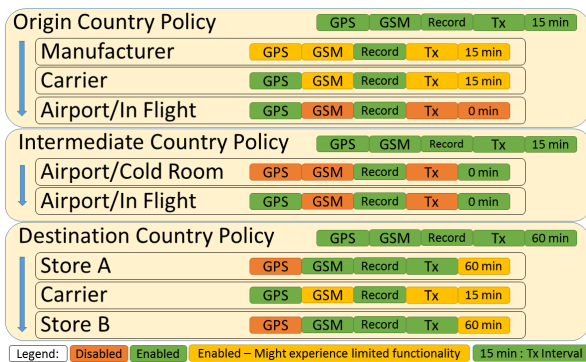


Figure 3: Sample constraints and restrictions along a typical vaccine shipment path. These may be used to define the operational modes of the application.

The final destination of the data is a cloud storage appliance, built on the Amazon Web Services (AWS) [22] and Amazon Elastic Block Store (Amazon EBS) for persistent storage. The appliance stores the received data on a mongoDB [23] database with a Node.js [24] front-end. This configuration is both high performing, and can scale-out quickly to handle large volumes of realtime traffic. Equally importantly, the appliance can easily be configured



Figure 4: Deployable device hardware. From left to right: Battery, IoT platform with added SD card and accelerometer, and thermal probe.

to export data (batch or stream mode) to specialised domain tools, such as the vaccine chain modelling and simulation tools developed by HERMES [25].

In the rest of this section, we present our design of the adaptive continuous sensing application, as well as describing its function and goals, we highlight some of the applicable design choices, and map these to our implementation.

3.3 Hardware Platform

We utilise the u-blox C027 [26] as an application platform. The device is distributed on the mbed platform [27] which provides fast prototyping tools for ARM Cortex-M devices, including an RTOS and runtime environment, software development kit, libraries and drivers for peripherals such as networking, etc.

The platform (see Figure 4) has a Cortex-M3 microprocessor operating at 96MHz with 64KB of RAM and 512KB of FLASH memory. It features an on-board MAX-7Q GPS/GNSS receiver and a SARA cellular module, including external active GNSS antenna and cellular antenna. In addition, there are on-board Ethernet and CAN interfaces and 22 GPIOs for easy expansion, including SPI, I2C, UART and I2C to facilitate expansion for applications [26]. For our application, we expanded the basic platform with a HT Micro SD card reader, a 4GB SD Card, an MPU 9150 IMU and a DS18B20 Digital Temperature Sensor in a thermal probe style waterproof enclosure.

The device is powered by common 9V batteries. The hardware components that can be switched off and are capable of entering very low power states that require milliamps to microamps of current to operate.

3.4 System Architecture

To support flexible control, we propose to utilise an event-driven approach, in which events (monitored conditions) enable the device to switch between individual modes of operation (for example from a library of available behaviours). To configure a particular application, users specify a composition of their desired operating modes, which are mapped to events and behaviours. To realise this aim, we first need to define a method to encapsulate the operations of the device; these may refer to individual modules, such as the GPS radio, their functions, such as enabling/disabling a feature, or to complex arrangements of individual functions such as running in a power saving mode that involves the coordination of several

hardware and software functions.

Given such encapsulation, we can use composition tools, such as a Domain Specific Language (DSL), to create configurations that arrange several behaviours (under some set of well defined conditions) to describe the desired modes of operation of the device. Finally, configurations may then be verified statically, for example by a model checker, to ensure that the application will not enter undesirable states.

Behaviours.

The basis of our architecture is the capacity to alter the mode of operation of a device in response to changing events. Modes of operation define the specific tasks to be performed by an application; for example, having the GSM radio disabled when in flight. See Figure 3 for examples of modes. Modes of operation are defined as groupings of *events* that may be encountered by the device, and the *behaviours* it is expected to execute in response. Such events represent the conditions that must be true in order to run some set of given functions, and so describe how the available device resources may be utilised.

Composing application modes.

Figure 3 illustrates how we may decompose the tasks of a continuous monitoring application to a set of modes of operation, and then associate with each mode i) a triggering event (its location) and ii) a set of device behaviours. Table 2, shows an example of a GPS geo-fencing polygon defined in combination with the Mobile Country Code (MCC) from the GSM. The location events in turn go on to enable or disable various hardware modules or some of their specific functions.

The information in Table 2 can be composed in a simple Domain Specific Language (DSL) that defines sets of independent modes of operation, and the events that trigger them. Events are defined as boolean operations on the variables made available for monitoring by the device/application, such as sensor readings, and behaviours specify the logic to execute. Finally, the event loop may be defined as a simple while loop, or as a more complex modular event runtime with asynchronous notifications by, for example, using libevent [28].

Event			behaviours (Enable/Disable)				
Polygon	OP	MCC	GPS	GSM	Rec	Transmit	Sched.
(51.5,0.0) (51.5,0.3) (51.3,0.3) (51.3,0.0)	or	234	1	1	1	1	15 (min)
(50.1,0.3) (50.9,22.3) (43.1,24.9) (41.9,4.4)	or	208 222	0	0	1	0	60 (min)
(11.4,5.2) (11.3,12.4) (7.1,5.2) (6.7,9.2)	or	621	1	1	1	0	60 (min)

Table 2: Location Triggers and corresponding Modes of Operation. The device while in a geographic polygon or a Mobile Country Code enables (1) or disables (0) components accordingly.

Validating compositions.

It is, for example, possible for a sequence of events to result in the system switching off all devices that are capable of providing localisation information. In such a case, no further geographically-defined events can be recognised. To ensure that the application

functions as desired, it is helpful to define a policy that expresses desirable and undesirable states, and to ensure that the operation of the system cannot result in an undesirable state. Model checking [29], specifically Linear Temporal Logic (LTL) model checking, is well suited to this problem of validating the operation of finite state concurrent systems such as event-based systems. LTL enables us to define propositions with temporal properties, which can then be validated by the model checker.

A model formally describes the application’s modes of operation, events and behaviours, which are expressed using process types. All possible interactions between such events and behaviours will be explored exhaustively in order to prove that the system functions as desired according to the allowed modes of operation and cannot become inoperable.

For example, the first mode in the Table 2 can be validated using the following LTL formula. Assuming polygon A and B are the specified areas and \Box roughly translates to *always*, while \Diamond is *eventually*, we have:

$$\Box(in_polygon_A|mcc = 234 \Rightarrow \Diamond gsm_on \& gprs_on \& recorder_on \& transmit_on)$$

When model-checked, each active process is interleaved with one another, so that all possible states are visited whilst checking for the claims specified in LTL. If a claim is not satisfiable by a given model, a counterexample will be generated.

Runtime.

Given an input (the configuration of an application), the function of the DSL is to validate it and then support the execution of the actions defined. This may be achieved in two ways: first, the DSL may validate and then compile the configuration to an executable, and then load the executable onto the device. This method offers the capacity to optimise the final executable for a device, and is suitable for specialised scenarios, such as when memory is at a premium, but limits the flexibility of the final application. The runtime in this case may, for example, be a minimalist interpreter (virtual machine) on the device built using the LLVM [30] compiler suite, which executes the byte code of the DSL. In the second case, we may execute the DSL directly on the device, in which case we can be more flexible with how we deploy application to the device, but possibly at the cost of requiring more resources. In the simplest case, this can be achieved by using, say, the ANTRL suite [31] to parse the DSL and generate configurations that are executed on the device.

To ensure the liveness of the system, we require that behaviours are non blocking or, at least, are guaranteed to return eventually. For responsiveness, once a behaviour returns, the last action executed by the event statement must be to check whether any new conditions have arisen since the last call. Finally, being adaptive to the environment means that it is the very likely that applications run into undefined modes. For this reason, behaviours may define recovery routines that state how to handle failure. Recovery code may be run by independently-scheduled watchdog routines to ensure that applications behave as expected or may be handled directly by the components that raise/receive them.

To illustrate, when an enabled behaviour such as the GSM module is executed, it attempts to power-up the device, initialise it, collect cell tower information such as the MCC or the Location Area Code (LAC), power it down, and finally return the data it has collected in JSON format. If, on the other hand, the module is executed and is either not enabled or cannot fulfil a task, it will fail gracefully, notify the runtime of its failure conditions and be handled accordingly. For example, if the GSM module does not have

a network connection, the intended transmission is saved automatically, and will be tried again later. Finally, for each measurement, the application messages record the following: the GPS coordinates, GPS Time, On board clock time, Mobile Country Code, Mobile Network Code, Location Area Code, Temperature and Acceleration.

3.5 Implementation

Behaviours.

We implement simple behaviours to abstract away from the hardware components such as GPS, the GSM modem, SD Card, Accelerometer and Thermal probe. Behaviours are the software encapsulation of individual hardware components and their functionality. As well as exposing the functionality of modules, behaviours (implemented as classes) enable us to express complex functionality easily. For example, given that GPS measurements are taken infrequently, it is prudent to power down the module whenever possible. Using the behaviour abstraction we simply couple power on/off actions to the usage of the module to define a distinct power saving GPS mode.

Composing application modes.

The behaviour encapsulation enables us to define a unit of distinct behaviour flexibly. For example, we may define two distinct behaviours on the GSM module: i) scan environment, for localisation; and ii) send message, for reporting. We define different constraints on each behaviour. In this way, we can map the compositions of an application to sequences of enabled/disabled and atomic behaviours. Practically, we may realise this by associating each distinct behaviour with a Boolean enabled/disabled flag. This way, when we define a new mode of operation (an event and its corresponding behaviour), we are essentially stating which behaviours are enabled and may be used.

Table 2 illustrates how parts of the application policy given in Figure 3 may be mapped to a configuration specifying which behaviours are enabled and when. Here, each polygon defines a mode of operation, which enables/disables a set of scheduled behaviours. Note that, to provide redundancy for switching the operating mode of the device, the events in Table 2 each define two distinct conditions to trigger each mode.

Validating compositions.

We have defined a formal model and claims in LTL using Promela and Spin [32]. The model enables us to define process types for devices, such as the GPS, which dictate the conditions on them such as when they can or cannot be used, as depicted below.

```
proctype GPS()
{
  do
    .....
    if
      :: (mode != MODE_IN_FLIGHT) \
        -> hw_gps = true
      :: (mode == MODE_IN_FLIGHT) \
        -> skip
    fi
  od
}
```

Having specified a model for the application's behaviour, we can use LTL verification conditions in order to check that the restrictions and policies are met by the model. For instance:

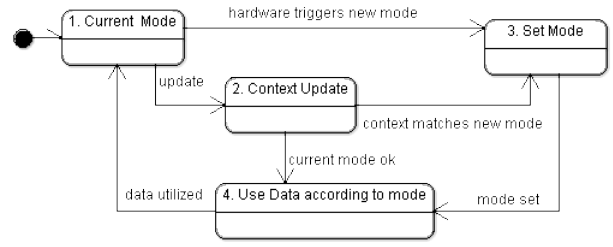


Figure 5: State Diagram depicting the current mode of operation, its alteration and the usage of data gathered.

$$p_1 = \{\Box(hw_{gsm} == true \wedge inPolygon == false \Rightarrow \Diamond(hw_{transmit} == true))\}$$

The above formula (p_1) specifies a policy that roughly translates to: “as long as GSM hardware is switched on and the current location is outside a given polygon, we will eventually transmit data”. It can be checked against the model of the application to ensure that it is not violated, i.e. no counter examples exist.

Using the LTL claims as exemplified above, we were able to discover errors in the application early and have some guarantees on its expected operation. Note that, though not explored in this work, using a DSL to specify the configuration of the application, also enables one to auto-generate the application, reducing the potential for error.

Runtime.

Figure 5 illustrates the operation of the application's run-time. This consists of a main-loop that runs continuously to detect for new event triggers, for example changes in location. Events are implemented as a background threads that read and set state variables. For example, from Table 2, to establish its current location, the device first attempts to check its GPS coordinates and, if those are not available, falls back to using GSM radio to obtain the current Mobile Country Code (MCC) or Location Area Code (LAC). If an event is true the corresponding mode is entered by setting its enabled behaviours, such as the required hardware for the mode. Finally, in our implementation, the runtime schedules modes as defined by their configuration and the application is run periodically.

Handling failure.

To handle failure, for example losing network coverage or a module failing, behaviours define recovery routines. In the simplest case, a recovery system can be built around simple notifications, for example failure in behaviours may be signalled by returning an error value or raising an exception. These can be handled by the runtime; for example, messages may be placed on a heap data structure for centralised handling or directly by receiving functions. Notifications enable behaviours to recover from failure explicitly: in enabling behaviour to define their own recovery routines, our goal is to provide an explicit way to pre-empt deadlock situations and guide the response of the application.

4. RELATED WORK

Over the years, significant effort has been devoted to the analysis and improvement of vaccine supply chains in an attempt to improve health and save lives in developing countries. It is increasingly being recognised that today's health problems in the developing world are tomorrow's problems in the developed world. For example, the

emergence and virulence of Ebola in Africa have spurred very rapid action in vaccine development in the developed world. There are existing efforts in the areas related to our problem domain, and, in this section we discuss some of the prominent prior work in the field.

4.1 The EVM and HERMES tools

Effective Vaccine Management (EVM) is a joint initiative by the World Health Organisation and UNICEF that was launched in 2010. It is a continuous quality-improvement process that aims to help countries improve and upgrade their immunisation supply chains by evaluating the current performance against best practice [33]. The EVM assessment tool compares supply chains against the standards of the industrialised world to generate an overall score in nine indicative areas: vaccine arrival, temperature control, storage capacity, infrastructure maintenance, stock management, distribution, vaccine management and information systems as described and analysed further in [33]. EVM employs a three-step improvement process which begins from assessment, produces a plan for recommended actions, that will finally be implemented. It is of key importance that a device capable of effectively providing data for as many as possible of the above indicative areas could significantly assist and accelerate the analysis of vaccine supply chains.

Another important tool towards effective immunisation is the Highly Extensible Resource for Modelling Event-Driven Supply Chains (HERMES)[25]. HERMES is a platform that “allows users to generate a detailed discrete event simulation model of any vaccine supply chain” [25]. By providing data about the a given supply chain with respect to its structure, storage locations, type and capacity, and, if available, up to the detail of demand for and usage of individual vaccine vials, the impact of introducing different approaches in storage and monitoring can be examined. In addition, the reorganisation of every aspect of a particular supply chain can be attempted in the model, allowing all the aspects of its performance to be examined. HERMES can accept data at fine granularity, to model every storage and immunisation location, storage device, transport device, vaccine vial etc. in the supply chain [25].

4.2 Project Optimize

In 2011, project Optimize [34], in collaboration with the Albanian Ministry of Health, installed an SMS-based temperature logging and alarm system that has been sending notifications to health workers and supervisors in charge of storage locations [35]. The results are encouraging and have successfully identified faults in the supply chain. It is reported that as a consequence, supervisors were more keen to take appropriate follow-up measures, procedural compliance has increased and equipment failure rates were exposed [36].

Project Optimize also tested FoneAstra [6], a continuous temperature monitoring system that utilises microcontroller-based devices attached to mobile phones. The microcontroller is equipped with a thermometer and automatically records temperature which together with the mobile handset’s cell tower ID information is transmitted to a server via SMS. FoneAstra had success worth mentioning in approximating test routes in India and was also used to examine possible benefits to the cold chain equipment management in the Albania study, where the device was used to raise SMS alerts if refrigerator temperatures shifted out of acceptable ranges.

Further work within Project Optimize has evaluated ocean freight as an alternative and economic way of transporting vaccines. [37]. Due to the increasing costs of airfreight, and as the technology of the actively-cooled Reefer Containers enables them to be remotely monitored and adequately controlled, this op-

tion of transporting vaccines might be expected to enter supply chains. An example of satellite tracking of Reefer Containers is the SMARTemp [38] which utilises Iridium satellites and is capable of sending alerts via email or SMS to cargo teams, should corrective action need to be taken. It is also worth mentioning the installation of GSM cell towers in 80% of Maersk Line Vessels since 2012 to facilitate real time communication needs required by future applications [39]. Another collaboration between Project Optimize and ColdCloud [40] has examined cloud-based solutions for monitoring refrigeration equipment in real time, providing a portal-based overview of the cold chain with escalating user-defined alerts via SMS and email.

5. CONCLUSIONS

WHO and UNICEF jointly report that out of the 65 low and lower-middle income countries assessed on effective vaccine management in 2013, none of them has met the recommended WHO standard [33]. This indicates the urgent need for high quality data from all parts of the vaccine supply chains and shows that there is huge room for improvement. Although a large variety of monitoring devices are deployed today, few are capable of functioning across the supply chain and supporting real time remote monitoring. Of those, we have found none that implement dynamically configurable and verifiable continuous sensing.

In this paper, we presented an approach to building an adaptive continuous monitoring application. Built on an easily configurable and extensible sensing platform, the application is able to support multiple modes of operation and to switch between them easily. The application can be deployed for both long-term monitoring or to track supplies, and can supply the operator with real-time information. Using our model-checking approach, it is possible to ensure that an application intended for deployment adheres to regulatory constraints, and is robust to changes in context. Our prototype has enabled us to explore some of the problems that will need to be addressed in order to realise the WHO’s continuous monitoring vision for immunisation supply systems in 2020 [20].

While there is still a long way to go to realise fully adaptive continuous sensing applications that can be deployed reliably, such applications could bring real gains. They offer the potential to improve the volume and quality of the risk assessment data that is currently gathered throughout the vaccine supply chain. Moreover, the use of standardised hardware provides the means by which a large category of currently deployed monitoring appliances can be replaced with a standardised solution.

6. REFERENCES

- [1] U. N. Foundation, “What We Do: Global Health,” tech. rep., UN, 2013. <http://www.unfoundation.org/what-we-do/issues/global-health/> Accessed: 10/09/2014.
- [2] GAVI, “Gavi’s mission.” <http://www.gavi.org/about/mission/>. Accessed: 16/09/2014.
- [3] Umit Kartoglu and Julie Milstien, “Tools and approaches to ensure quality of vaccines throughout the cold chain,” *Expert Rev. Vaccines*, vol. 13(7), 2014.
- [4] W. H. Organization, “TEMPERATURE MONITORING DEVICES : AN OVERVIEW,” tech. rep., WHO, 2013. <http://tinyurl.com/ky58f9x> Accessed: 29/07/2014.
- [5] D. T. M. JC and B. G., “Cold chain management: Knowledge and practices in primary health care facilities in Niassa,

- Mozambique,” *Ethiop J Health Dev*, vol. 21(2), 2007.
- [6] R. Chaudhri, E. O’Rourke, S. McGuire, G. Borriello, and R. Anderson, “Foneastra: Enabling remote monitoring of vaccine cold-chains using commodity mobile phones,” in *Proceedings of the First ACM Symposium on Computing for Development*, ACM DEV ’10, (New York, NY, USA), pp. 14:1–14:9, ACM, 2010.
 - [7] European Union, “Good Distribution Practice of Medicinal Products for Human Use (2013/C 68/01),” 2013. <http://tinyurl.com/cab8t7q> Accessed: 09/09/2014.
 - [8] World Health Organization (WHO), “Study protocol for temperature monitoring in the vaccine cold chain.” http://whqlibdoc.who.int/hq/2011/WHO_IVB_05.01_Rev.1_eng.pdf, 2005. Accessed: 09/09/2014.
 - [9] D. M. Matthias, J. Robertson, M. M. Garrison, S. Newland, and C. Nelson, “Freezing temperatures in the vaccine cold chain: A systematic review,” *Vaccine*, vol. 25, 2007.
 - [10] Q. Ren, H. Xiong, Y. Li, R. Xu, and C. Zhu, “Evaluation of an Outside-the-Cold-Chain Vaccine Delivery Strategy in Remote Regions of Western China,” *Public Health*, vol. 124(5), 2009.
 - [11] T. S. P. Varinsathien, A. Rasdjarmrearnsook, and P. Tharmaphornpilas, “Exposure to heat and freezing in the vaccine cold chain in Thailand,” *Vaccine*, vol. 25, 2007.
 - [12] T. Wirkas, S. Toikilik, N. Miller, C. Morgan, and C. Clements, “A vaccine cold chain freezing study in PNG highlights technology needs for hot climate countries,” *Vaccine*, vol. 25, 2007.
 - [13] A. K. Naik, M. P. Rupani, and R. K. Bansal, “Evaluation of Vaccine Cold Chain in Urban Health Centers of Municipal Corporation of Surat City, Western India,” *Int J Prev Med.*, vol. 4(12), pp. 1395 – 1401, 2013.
 - [14] “Temperature monitoring devices performance specifications and verification protocols.” <http://tinyurl.com/kq2apsb>, 2014. Accessed: 10/09/2014.
 - [15] United States Pharmacopeia, “Monitoring devices-time, temperature and humidity.” <http://tinyurl.com/px54p7o>, 2009. Accessed: 09/09/2014.
 - [16] Andrea Gruber, “Air transport logistics for time- and temperature-sensitive healthcare products,” tech. rep., Pharmaceutical Outsourcing, 2012. <http://tinyurl.com/mklu7ac> Accessed: 09/09/2014.
 - [17] W. H. Organization, “OPTIMIZE Senegal Report,” tech. rep., PATH, 2013. <http://tinyurl.com/or5hekf> Accessed: 04/07/2014.
 - [18] W. H. Organization, “Mid-Level Management Course for EPI Managers Block III : LOGISTICS Cold chain management.” <http://tinyurl.com/pbfg869> Accessed: 02/09/2014, 2005.
 - [19] W. H. Organization, “Vaccine storage - time / temperature,” in *Procurement of vaccines for public-sector programmes A reference manual*, pp. 750–980, 1997.
 - [20] W. H. Organization, “Developing a Vision for Immunization Supply Systems in 2020 Landscape analysis summaries,” tech. rep., PATH, 2011. <http://tinyurl.com/omfq8m5> Accessed: 04/07/2014.
 - [21] Netfuse, “Netfuse.” <http://www.netfuse.net/>. Accessed: 10/09/2014.
 - [22] Amazon.com, “Amazon Web Services (AWS) Cloud Computing Services.” <http://aws.amazon.com/>. Accessed: 28/08/2014.
 - [23] mongoDB, “mongoDB.” <http://www.mongodb.org/>. Accessed: 02/06/2014.
 - [24] node.js, “Node.js.” <http://www.nodejs.org/>. Accessed: 01/06/2014.
 - [25] Hermes, “What is the Vision of HERMES ?.” <http://hermes.psc.edu/vision.html>. Accessed: 02/09/2014.
 - [26] Ublox, “Ublox mbed enabled Internet of Things (IoT) platform.” <http://www.u-blox.com/en/c027-internet-of-things-starter-kit.html>. Accessed: 10/09/2014.
 - [27] mbed, “Development platform for devices | mbed.” <http://mbed.org/>. Accessed: 10/09/2014.
 - [28] N. Mathewson and N. Provos, “libevent.” <http://libevent.org/>. Accessed: 10/09/2014.
 - [29] D. P. E.M. Clarke, O. Grumberg, *Model Checking*. MIT Press, 1999.
 - [30] LLVM Developer Group, “LLVM.” <http://llvm.org/>. Accessed: 10/09/2014.
 - [31] ANTLR, “ANTLR.” <http://www.antlr.org/>. Accessed: 10/09/2014.
 - [32] G. Holzman, “ Verifying Multi-threaded Software with Spin.” <http://spinroot.com/spin/whatispin.html>. Accessed: 27/09/2014.
 - [33] W. H. Organization, “Effective Vaccine Management,” 2014. <http://tinyurl.com/lnopn42> Accessed: 10/10/2014.
 - [34] PATH, “Project Optimize.” <http://sites.path.org/vpsse/optimize/>. Accessed: 02/09/2014.
 - [35] W. H. Organization, “Albania : Demonstrating the benefits of an online immunization registry,” tech. rep., PATH, 2012. http://www.path.org/publications/files/TS_opt_albania.pdf Accessed: 10/07/2014.
 - [36] W. H. Organization, “Temperature monitoring for vaccine quality,” tech. rep., WHO, 2011. http://www.path.org/publications/files/TS_opt_temp_monitor.pdf Accessed: 10/07/2014.
 - [37] W. H. Organization, “Ocean Shipment of Vaccines,” tech. rep., PATH, 2013. http://www.path.org/publications/files/TS_opt_ocean_ship.pdf Accessed: 10/07/2014.
 - [38] APL, “A first from APL: satellite tracking of container temperatures.” https://aplinfo.apl.com/press_releases/html/press_release_20100608.html. Accessed: 02/09/2014.
 - [39] Maersk Line, “Maersk Line installs masts without sails! (They are GSM...)” <http://tinyurl.com/o6n7k82>. Accessed: 02/09/2014.
 - [40] ColdCloud, “ColdCloud.” <http://www.coldcloud.co.za/what-is-it>. Accessed: 16/09/2014.