



An Internet of Things (IoT) Model for Optimising Downtime Management: A Smart Lighting Case Study

Brenda Scholtz¹ , Mando Kapeso¹, and Jean-Paul Van Belle² 

¹ Nelson Mandela University, Port Elizabeth, South Africa
{brenda.scholtz, mando.kapeso}@mandela.ac.za

² University of Cape Town, Cape Town, South Africa
jean-paul.vanbelle@uct.ac.za

Abstract. In today's global, competitive economy, downtime has been identified as a key performance indicator for field service organisations. The emergence of an Internet of Things (IoT) has brought new enhancement possibilities to various industries such as the manufacturing and field service industry. This paper provides a vision and motivation for using IoT in Field Service Management (FSM) in order to address data quality and service delivery issues. The theory of information quality was used to undergird the research and a model for the optimisation of downtime management in the field service industry using the IoT is proposed. The model was used to drive the design of a "proof of concept" prototype, the KapCha prototype. The paper also includes a report on an empirical study of the application of the proposed IoT model in FSM. The experiment findings showed that the prototype reduced the round trip delay time for sending and receiving data and was scalable. As a result, access to quality information supporting advanced data analytics and artificial intelligence was provided. Therefore, service technicians can be alerted more quickly as soon as any potential technical problems occur. In turn improved diagnostics and more efficient decision making can be achieved. The model and the lessons learned provide valuable guidance to other researchers and fill the gap in research of empirical studies conducted on IoT implementations.

Keywords: IoT · Smart lightning · Design science research

1 Introduction

The lack of accurate, available and real-time information is a common challenge faced by field service organisations [1]. As a result downtime, which is an important performance measure in this domain, is negatively impacted. Downtime is defined as the time between a customer's request for service and the completion of the service by the field service team to rectify the problem [2]. Field service management (FSM) refers to the support provided by hardware and software in the management of field service operations and involves the management of the activities and processes that are associated with field services. There is a need for solutions that efficiently address the challenges of FSM and downtime management and that support the provision of

quality information and ultimately improved decision making and service delivery levels [3].

One application of downtime management in the field service domain is that of the maintenance of smart lights. Smart lighting projects have been undertaken by municipalities as a result of a drive for improved energy management within cities [4]. In the context of cities, streetlights are one of the most important assets to maintain as they provide safe roads and enhanced security for homes, businesses and city centres. However, they are costly to operate and account for an estimated 40% of the amount of electricity spent in an urban city [5]. To address this issue, city managers are implementing smart lighting solutions. Smart lighting consists of heterogeneous and multidisciplinary areas of lighting management, with the possibilities of integrating a wide range of sensory and control technologies with ICTs. This integration can improve efficiencies in lighting products and lower the negative impact derived from the use of energy for illumination. Smart lighting provides intelligent features and interfaces for lighting solutions in the ambient, commercial and public domains [4, 6]. Smart lighting is linked to the concept of a smart city, which is an urban development that envisions the efficient management of a cities resources and services with the use of integrated ICT solutions [4]. Smart cities play an important role in the sustainable economic development of countries or states seeking to attain environmental sustainability. Smart cities are made possible through the abundance of smart devices, smart objects and the emergence and rapid growth of technologies such as the Internet of Things (IoT). The IoT is described as a decentralised system of “smart” objects with sensing, processing, and network capabilities [7].

Extensive research has been conducted related to the IoT [16, 18, 26]. In particular several studies have proposed various architectures for IoT, such as a general reference architecture [27] and others in certain domains such as smart metering [28]. However, limited studies can be found that report on empirical studies of IoT applications in practice and findings and lessons learnt from these applications. There is a need for research into how technologies in the IoT can be applied to various business domains [20].

This paper addresses this gap by investigating an IoT application in the domain of smart lighting. The purpose of this paper is to propose an IoT model that addresses the challenges of information quality leading to poor downtime management. The paper reports on the application of this model in the smart lighting domain. The model includes IoT compatible technologies and techniques (protocols and formats) to support successful downtime management. To address this purpose, a critical analysis of the literature related to FSM, downtime management and IoT was conducted (Sect. 2). The context was a smart lighting organisation in South Africa (Sect. 3). From the literature and consideration of the context, a theoretical model was derived (Sect. 4). The model was used to design the architecture of and to implement a prototype for the case study (Sect. 5). The experiments conducted revealed that the new architecture and protocols implemented resulted in a lower Round Trip Delay time and was scalable (Sect. 6). The quality of information was improved and provided a foundation for advanced data analytics and artificial intelligence (AI), since the system provided intelligent information to technicians and managers; thereby improving diagnostic decision making, downtime management and service delivery.

There are several contributions and implications for future research that are identified from this study (Sect. 7). The practical contribution is the model, which can provide guidance to practitioners working in the field service domain and for system designers. On a theoretical level the model and the implementation issues identified contribute to the body of knowledge regarding the application of IoT models, architectures and network protocols.

2 Literature Review

2.1 Challenges in Field Service Management (FSM) and Downtime Management

In a competitive global economy where every organisation is looking at ways to cut costs, increase efficiency and gain a competitive advantage, organisations have become more customer-centric. The effectiveness of field services provided by technicians affects everything from the retention of customers and the profitability of the organisation [1, 8]. With field-based services, customers receive either an on-site or a remote service [2]. FSM operations include tracking vehicles, scheduling and dispatching employees, and integration of these operations with a back-office system for inventory, logistics and marketing. FSM includes elements such as Enterprise Asset Management (EAM), maintenance support, sensor networks, Radio Frequency Identification (RFID) tags, technical support, contract management and product life-cycle management. The FSM market has seen a steady growth and evolution in the last 10 years [9], which can be attributed to new technology developments, as technology is a driver in improved after-sales service innovations.

Downtime management is an important measure of performance for field services for both the organisation providing the service and the customer [10]. From the customer's perspective, that is the organisation undergoing downtime, the downtime period has operational implications such as reduced productivity levels and delayed delivery of services to the organisation's clientele. It is therefore imperative that downtime is kept to a minimal period. Service providers have to adequately manage downtime in order to satisfy its customers and by doing so efficiently they may gain a competitive advantage. Agnihotri [2] classified downtime into two subcategories, response time and on-site time. Response time is the time between the customer's request and the service team's arrival on-site. On-site time is the duration of time taken between the service team's arrival at the customer's site and the rectification of the problem. Corrective maintenance occurs when the machinery breaks down and includes activities undertaken to diagnose and rectify a fault so that the failed machine, equipment or system can be restored to its normal operational state, thus reducing the extent of downtime.

A lack of information related to a technical breakdown can result in longer cycle times and possibly a second service visit, thus resulting in longer periods of downtime for customers [8]. A malfunctioning piece of industrial machinery on a manufacturing floor can translate into tens of thousands of dollars per minute. It is important to make critical information immediately available to field technicians and management with

high levels of accuracy. Critical data related to the problem must be accurate, available anywhere, and dynamically changing along with the day-to-day operations of field service teams. Access to this information can assist with optimising the problem detection step in FSM and field service providers can determine strategies to ensure that downtime is minimised and managed with optimal efficiency. Within IS literature, information quality (IQ) can be used as a dimension of IS success [12]. Knowledge is functionally related to data and information, thus it follows a hierarchy (data → information → knowledge), termed as the knowledge hierarchy [11]. Our study classified the problems in FSM related to information that impact downtime management according to six of the attributes of IQ proposed by [12]. These are:

- Timeliness: lack of access to real-time information [1];
- Completeness: missing information [1, 3, 8];
- Accuracy: inaccurate information [1];
- Relevance: aggregated or de-aggregated information [14, 15];
- Consistency: lack of integration between enterprise and FSM systems [14].

This analysis also confirmed the findings of [13] showing a significant relationship between IQ and individual impact. Individual impact is measured in terms of decision-making performance, job effectiveness, and quality of work. Challenges faced by FSM organisations with regards to IQ resulted in negative impact on decision making and service delivery. Inaccurate or missing information and a lack of real-time availability of information to employees onsite in the field (for example dispatchers and service technicians) resulted in operational challenges [1, 3, 8]. Information related to the customer or the equipment under maintenance or repair is not always readily available to field service employees, resulting in the poor scheduling of field employees, the ineffective management of field service resources such as service parts [3] and ultimately in poor service delivery. In a study by Lehtonen [1], it was reported that service teams could not provide a service due to missing spare parts. The main reason for this was the inaccurate information on the spare parts that was taken to the client at the time of repair. Challenges in FSM within Enterprise Systems may also arise due to the lack of accessibility and integration of various systems [14]. For example, geographical data is found in Geographical Information Systems (GIS), whilst maintenance-related data and reports are often stored in an Enterprise Resource Planning (ERP) system, thus resulting in integration and consistency issues. Schneider [14] reported issues related to the use of aggregated data within an ERP system. For example, in an ERP system electricity usage data for a manufacturing plant is usually stored as an aggregated figure for all work centres within the plant. Aggregated data makes the operational performance monitoring of a single work centre or equipment within the plant difficult.

Access to real-time information aids organisations in optimising FSM since it can minimise the time for the service team to locate a client location by using GPS services and can reduce the on-site time spent servicing a clients' request [1, 2]. Real-time access to the clients' location eliminates the need for the service team to return to the service provider's facilities in order to get information about a new client's request, thereby optimising the scheduling element [3].

2.2 Applications of the IoT

The IoT has brought new functionality possibilities for many industries such as manufacturing and field services [16, 17]. It is expected that soon more than fifty billion devices ranging from smart phones, laptops, sensors and game consoles will be connected to the Internet through heterogeneous access network technologies [18]. However, the successful implementation of an IoT system introduces several other challenges. The abundance of data provided by sensors can introduce inefficiencies in data transfer and a need for aggregated data since sensor nodes are constrained by limited resources, for example computational power, memory, storage, communication, and battery energy [15]. These constraints provide an important challenge to design and develop approaches to information processing and aggregation that are efficient and make effective use of the data. For a given query, it may not be necessary or efficient to return all the raw data collected from every sensor – alternatively information should be processed and aggregated within the network and only processed and aggregated information returned. From a system level perspective, the IoT can be viewed as a dynamic, radically distributed, networked system, consisting of many smart objects that produce and consume information [19]. It can optimise business processes by leveraging on advanced analytics techniques applied to IoT data streams [19]. Thus, it provides good potential for addressing the downtime problem, if successfully implemented.

Although technology advances enable the possibility of the IoT, it is the application of the IoT which is driving its evolution [18]. The potential social, environmental and economic impact that the IoT has on the decisions we make and the actions we take is its main driving force. For example, having accurate information about the status, location and identity of things which are part of our environment opens the way for making smarter decisions. The application domains that the IoT includes can provide a competitive advantage beyond current solutions. In its inception the IoT was used in the context of supply chain management with RFID tags as the enabling technology [7]. However, in the past decade its applications have covered a wide range of industries, including transportation and utilities, to name just a few. Hwang et al. [20] classified the potential business contexts of IoT into three different factors: industry applications (for example government, education and finance); service domains (for example transportation, asset management) and value chain activities (for example sales and marketing, service or procurement). On the other hand, Borgia et al. [18], classified the IoT into three application areas: industrial (for example agriculture, logistics or other industrial applications), health/well-being and smart city. The smart city factor includes safety, mobility, buildings, road conditions, waste collection and public lighting.

3 Context of Research: Smart Lighting

The case study used in this research is a smart lighting system that is maintained at an engineering consulting and research organisation in South Africa. For purposes of anonymity, the organisation will be called LightCo. The smart lights that are used as

outdoor luminous equipment for parking bays and security lights for building facilities and are grid independent; meaning they are not connected to a local or municipal electricity provider for the energy needed to light them. An interview was conducted with one of the senior engineers at LightCo in order to establish an overview of the environment as well as the challenges faced by the organisation in delivering maintenance services for the smart lighting environment.

Smart lighting consists of the integration of intelligent functionalities and interfaces at four complementary levels [4], namely: the embedded level; system level; grid level and communication and sensing level. The embedded level is the lighting engine or the light itself, whilst the system level is the luminaries and lighting systems. The grid level consists of the management and monitoring of the power sources, energy generation and plants and the distribution of utilities and appliances. The final level is the communication and sensing level, which provides complete lighting solutions with monitoring, control and management of the applications.

The smart light unit at LightCo consists of an on board 48-voltage battery pack that is used as an energy storage unit. The solar panel is used to harness solar energy and the wind turbine generates electricity by the turning of a generator. The architecture of the smart lighting system allows for remote monitoring. The smart light also contains sensors and actuators that enable it to measure environmental variables and to respond to specific conditions by means of the actuators. The sensors include ambient sensors on the solar panel and voltage and current sensors on the circuit board of the smart light. Furthermore, the smart light is uniquely identifiable and contains on-board microcontrollers that provide computational and communication capabilities. The microcontrollers receive voltage and current data readings from the solar panel and wind turbine and also record the voltage and current that is outputted to the LED light. The battery management system manages the flow of current to the battery. Once these readings have been recorded they are then sent to a remote server for processing.

Prior to starting this study, the smart lighting system at LightCo did not provide for efficient or effective downtime management. Technical problems with the lights were not being reported timeously and were not correctly diagnosed due to IQ issues reported in literature [1, 14]. These problems could be for example, an LED light or circuit board is damaged. The system that was in place for detecting technical problems with their lights used a Global System for Mobile Communications (GSM) SMS-based messaging/polling protocol to transfer data from a smart light to a server at a remote location. This protocol was reported as inefficient due to its high latency times and high data costs affiliated with the sending and receiving of SMS messages. Increasing the latency was not an option, since it would increase the data costs. Data transmission was not bi-directional and data was merely recorded in a CSV file, with no processing performed on it. An Arduino microcontroller was situated in each smart light with a GSM Shield, which allowed the Arduino board to send and receive an SMS as well as connect to the Internet using the GSM library. However, the system did not use the GPRS wireless component that would enable the Arduino to connect to the Internet. Technicians had to manually peruse the data to diagnose any issues or potential issues.

4 IoT Model for Downtime Management

The Three Phase Data Flow Process model proposed by Borgia et al. [18] (Fig. 1), the four layers of IoT [25], and IQ theory were used as the main guiding theories for the proposed IoT Model for Downtime Management (Fig. 2). The model describes the flow of data in the IoT over three phases [18], namely the Collection Phase; the Transmission Phase; and the Process Management and Utilisation phase and four layers [25] (the Sensing Layer; the Networking Layer; the Service layer; and the Interface layer). The Sensing Layer consists of hardware that senses and controls the physical world and acquires data. Examples are RFID, sensors and actuators. The Network Layer provides networking support and transfers data over either a wireless or a wired network. The Service Layer is responsible for the provision of services to satisfy the user needs and creates and manages services. The Interface Layer (or Application Layer) interacts with other applications and users.

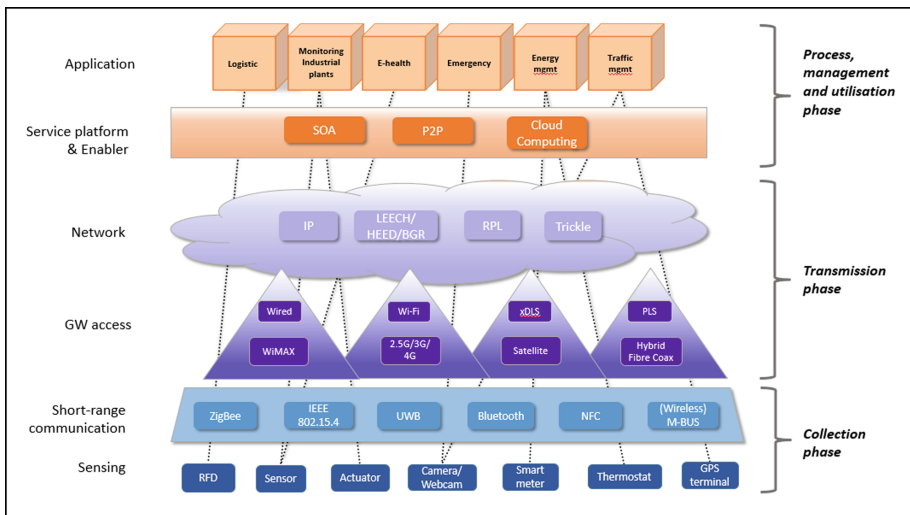


Fig. 1. The three phase data flow process [18]

The Collection Phase reports on the event driven processes during the collection and acquisition of the data from the environment [18]. Data acquisition technologies attached to sensors and cameras collect information about the physical environment (temperature, humidity and brightness), or about the objects (identify and state) in real-time; while data collection is accomplished by short range communications, which could be open source standard solutions or proprietary solutions. In the FSM context these would be integrated into the equipment or assets in the field, for example the smart light. The Transmission Phase involves mechanisms that deliver collected data to various applications and external servers [18]. Once data has been collected it must be transmitted across the network so that it can be consumed by applications. For wired

technologies the standard is Ethernet IEEE802.3. The primary advantage that wired networks have for data transmission is that they are robust and less vulnerable to errors and interference. However, they are costly. Therefore Wireless LAN (WLANs) are often used to access the network. Due to the flexibility of WLANs, it is believed that they will be the main communication paradigm of the IoT. However, the restricted wireless spectrum available for cellular networks is a major limitation to their widespread use.

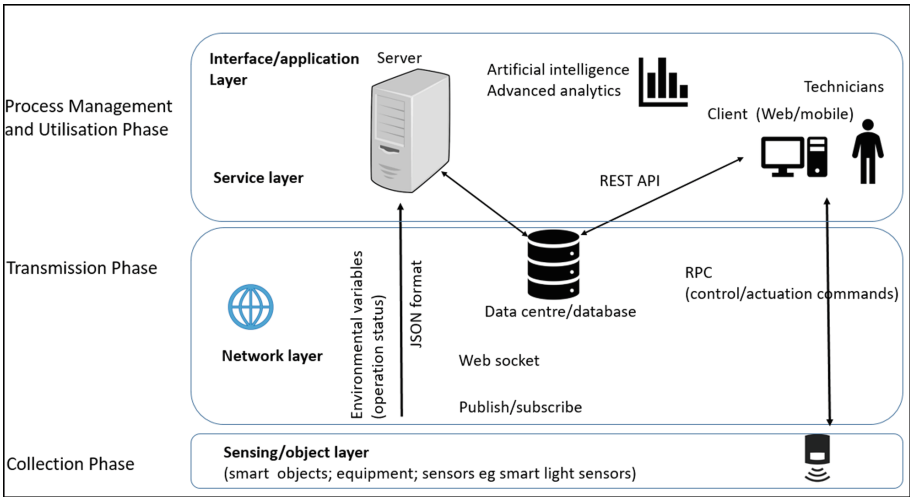


Fig. 2. IoT model for downtime management

The Processing, Management and Utilisation Phase incorporates the processing and analysing of information flows, data forwarding to services and applications and the provision of feedback to control applications [18]. It also involves device discovery and management, data filtering, aggregation and information utilisation. The Service Platform & Enabler sub-phase covers an important role for managing these functions and is necessary in order to hide the heterogeneity of hardware, software, data formats, technologies and communication protocols that are a key feature of the IoT. Its responsibility is to abstract all the features of objects, networks and services, and to provide a loose coupling of components.

5 Methodology and Development of the Prototypes

5.1 Methodology

The Design Science Research (DSR) methodology [21] was adopted in this study to create and evaluate the artefacts (model and prototype). The model was derived from a systematic literature review as well as from the case study of smart lighting

maintenance, which was used for implementing and evaluating the model. The Technical and Risk efficacy evaluation strategy from the Framework for Evaluation of Design Science (FEDS) is used in the DSR methodology for evaluations conducted in the design cycle of DSR and was used in this study to evaluate both the model and the KapCha prototype [22]. An artificial-summative evaluation was used to evaluate the design of the model, but due to space limitations these results are not reported on in this paper but are available on request. Iterative formative evaluations were conducted during the development of the prototype; after which a summative-naturalistic evaluation was conducted in order to determine the performance of the prototype under real-world conditions.

5.2 The Prototypes and Their Mapping to Requirements

The KapCha prototype was developed using an incremental prototyping process comprising of three prototype components (Table 1). **ProWebSoc** is the web socket protocol; **ProObjWeb** is the web socket client; and **ProDT** is the interface layer and web socket server. The IoT Model for Downtime Management (Fig. 2) was used to design the architecture of the prototypes.

Table 1. Prototype components

Collection and transmission phases		Process management and utilisation phase
ProWebSoc web socket protocol	ProObjWeb web socket client	ProDT (interface layer) web socket server
Data transmission protocols; security and standardisation	Transmission of data from the smart light and integration of CPS principles	Intelligent algorithms, advanced analytics and interfacing with applications and mobile technologies

Collection and Transmission Phases (ProWebSoc and ProObjWeb)

As an alternative to the SOAP/XML data transmission protocols used by LightCo prior to this intervention, a protocol based on JavaScript Object Notation (JSON) was implemented. JSON, is a text-based open standard format that is designed for human-readable data interchange and used for the serialisation of structured data making it easy for machines to parse and generate it. JSON is ideal for low processing computational capabilities (such as the smart light) and can result in less data that needs to be generated as compared to SOAP/XML.

ProObjWeb, through the web-socket client, enabled the smart light in the case study, as an OEM, to interface with a remote web-server using the KapCha web-socket protocol and to transmit data over a web socket protocol (**ProWebSoc**). Web sockets enable bi-directional communication (upstream and downstream) through the introduction of an interface and the definition of a full-duplex single communication channel that operates through a single socket [23]. They provide a reduction in network

traffic and latency as compared to polling and long-polling solutions that are used to simulate a full-duplex connection by maintaining two connections. They also reduce the amount of port openings on the server side, as compared to the traditional means of retrieving resources such as polling. This reduction also reduces maintenance of connection channels from the server side, therefore decreasing the overhead network traffic. The web socket protocol also has the ability to traverse firewalls and proxies, which is a problem for other protocols. The protocols provide real-time communication (RTC) between a smart object and a central system or other smart objects and supports ad hoc and continuous data transfer as well as operational status communication and Remote Procedure Calls (RPCs).

A GPRS wireless component was used to enable the Arduino to make use of web socket technology. The web socket client was developed on the Arduino board using the Arduino open source software and several web socket methods. During connection, the web socket detects the presence of a proxy server and automatically establishes a tunnel to pass through the proxy. The tunnel is established through the opening of a TCP/IP connection. The connection is established by the client issuing an HTTP connect statement to the proxy server for a specific host and port. Upon the tunnel being set up communication flows uninterrupted through the proxy.

The web socket protocol (**ProWebSoc**) was designed to work with existing web infrastructure, therefore the protocol specification defines that the web socket connection starts as an HTTP connection [24]. This guarantees full backwards compatibility with HTTP based communication protocols. The upgrade from an HTTP to web socket is referred to as a handshake. In this process the client sends a request to the server indicating that it wants to switch protocols from HTTP to Web sockets, by means of an upgrade header. During the handshake process the server accepts the request and responds with an upgrade switch header. The server acknowledges receipt of the client's request by taking the |Sec-Web socket-Key| value and concatenating it with a Globally Unique Identifier (GUID) in string form. An SHA-1 hash (160 bits) base64-encoded of this concatenation is then returned in the server's response. This prevents an attacker from tricking a web socket server by sending it carefully crafted packets using XMLHttpRequest or a form submission.

Web sockets are ideal due to the ability to use customised protocol calling depending on the service being offered [23]. In ProObjWeb, when the client receives a response with no errors the connection is upgraded to a web socket over the same TCP/IP connection. Once the connection is established data frames between clients and servers can be transferred. Once the web socket client application connects to the webserver, the webserver initiates an upgrade sequence to upgrade the connection from an HTTP connection to a web socket server. ProObjWeb was functionally tested using the web socket.org echo server, which allows developers of web socket applications to test the ability of their applications to successfully upgrade the connection from HTTP to a web socket protocol. The test results showed that ProObjWeb successfully managed to connect to the web.org server and upgrade from HTTP to web sockets.

Management and Utilisation Phase

The third prototype (**ProDT**) focused on the development of the web-socket server application, a decision tree algorithm implementation and a REST (Representational

State Transfer) API web interface. REST APIs with web socket Requests/Responses were used to form an intermediate layer between a client and the database, translate the raw data from the database to a format that the client requests and transmit the data. Most databases provide real-time notifications for added or updated data. However, real-time notification passage between the database and the client was required, therefore an Interface Layer was created in ProDT using a web socket server. The web-socket server also handled communication with the database, which is the core of the application architecture. These techniques (including the protocols) provided real-time notification between the database and the client, eliminating the need for the Ajax techniques of polling. The database contained information about each smart light such as the date of manufacture and installation, its GPS location, object data sent by the Smart Light, the fault issue result after data analysis and job card data.

In order to generate an issue, data analysis using a decision tree learning algorithm was implemented. A decision tree is a tree structure consisting of nodes that each represent a test of an attribute with each branch representing a result of the test [29]. The tree splits observations into mutually exclusive subgroups until observations can no longer be split. The ID4.5 is a popular splitting algorithm that builds a decision tree by employing a top-down, greedy search through the given sets of training data to test each attribute at every node. Decision trees require little effort for data preparation unlike some statistical techniques and they are easy to interpret. The data collected was categorised data and therefore is ideal for decision tree. Furthermore, as there was no historical information on the diagnosis of faults or issues the decision tree was the ideal AI technique to be used as the classifier would be developed from the expert's opinion of plausible issues. The diagnosis of a set of problems based on these opinions was determined as a classification problem.

The improvements in the quality of information provided by these techniques thus allowed for advanced data analytics and intelligent algorithms (such as decision trees) to be conducted on the IoT data streams. The ability to interface with mobile technologies was also provided.

6 Experiment Procedure and Findings

The aim of the experiments was to evaluate the Round-Trip Delay time (RTD) of messages, latency, accuracy of the decision tree analysis and scalability. Due to space constraints this paper only provides details of the RTD and accuracy experiments.

Round-Trip Delay time (RTD). The RTD experiment measures the time taken for a client to send a signal to a server and the time it takes for the server to acknowledge the signal and send a response [22]. In this context the client was the smart light and the server was the remote web socket server application. For the RTD experiment a connection had to be established between the smart light and the web socket server application. The experiment procedure involved running the applications in three cycles; the first cycle involved sending a data packet 10 times, then the second cycle 100 times and the third cycle 1000 times. The data packet was an array data object that was instantiated and sent to the server.

Two phases of experiments were performed: the first phase (local testing) consisted of running the server applications on a local host machine; and the second phase (remote testing) of the experiment involved running the server applications on a remote server. The performance metrics for the RTD evaluation were delay time and messages per second.

From the results of all three cycles it is evident that the KapCha web socket had a lower RTD time as compared to the Ajax protocol (Table 2). The results can be attributed to the fact that there are fewer HTTP overheads when using web sockets as compared to Ajax requests. Upon the connection being established all messages are sent over the single socket connection rather than the creation of new connections for new HTTP request and response calls created every time a message is sent over the Ajax protocol.

Furthermore, the web socket protocol had more messages sent per second as compared to that of the Ajax protocol. The messages per second for web sockets are higher because the web sockets establish the connection once over a single socket, unlike Ajax techniques that require multiple connections to be opened and closed during request/response calls. Therefore, web sockets do not have messages delayed during the connection process and can send more messages per second. The messages sent per second over the web socket protocol increased exponentially with the number of iterations completed.

Table 2. Experiment results – RDT and messages per second

First phase: local host					
Cycle	Num packets	Time (sec)		Messages	
		Ajax	Kapcha	Ajax	Kapcha
1	10	0.130	0.082	80.135	150.235
2	100	0.435	0.082	301.720	1502.235
3	1000	3.104	0.558	340.832	2215.621
Second phase: remote testing					
Cycle	Num packets	Time (sec)		Messages	
		Ajax	Kapcha	Ajax	Kapcha
1	10	1.241	3.705	3.212	1.752
2	100	22.016	17.213	2.907	3.352
3	1000	153.352	147.25	3.191	3.714

The results from the remote testing set of experiments revealed that the web socket protocol had a lower RTD time as compared to the Ajax protocol when the number of packets was lower than a certain level. This result could be due to the upgrade sequence overhead during the web socket handshake process. The additional overhead connection, however, is not significant as the number of iterations increase due to the maintenance of the single socket connection. The RDT results highlighted the advantages of applications that use web sockets have over HTTP polling mechanisms.

The advantages are lower latency and the provision of a single socket connection that enables the web server to push data to the client at will, creating a fully duplex bi-directional data exchange web-protocol.

Accuracy of decision tree: Prior to the development of the prototype there wasn't any data stored regarding the cause of a fault or the documentation of the diagnosis of a fault at LightCo. Therefore, the accuracy of the training dataset created was based on the expert's verification. The C4.5 decision tree algorithm was used to analyse the data and deduce the cause of the faults that occurred. For the experiment, the algorithm was deployed/executed to analyse three sample data set sizes of 50, 100 and 175. The number of correct predictions after each execution was recorded and verified by an expert at LightCo. This process was undertaken to establish the accuracy of the algorithm in diagnosing faults. The execution time of the algorithm was also recorded to determine the turnaround time of the fault diagnosis. The formula used for determining the accuracy percentage was:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} = \frac{f_{11} + f_{100}}{f_{11} + f_{10} + f_{01} + f_{00}}. \quad (1)$$

The accuracy results are summarised in Table 3. The sample size of fifty ($n = 50$) resulted in a percentage accuracy of 82.2% meaning that 41 out of the 50 predictions were correct. The sample size of 100 had 79 correctly predicted faults with a percentage accuracy of 79%. The final sample size had an accuracy percentage of 77% that is 96 faults were correctly predicted. Whilst the accuracy results were all above 70% additional testing is required to determine accuracy of larger datasets. However, this could not be done since previous records were non-existent and the training set was small. This is a limitation of this study. Future studies should perform the accuracy tests on the larger data set, which will increase rapidly with time. However, in spite of this limitation useful results and lessons learned were obtained regarding the IoT techniques used in the model.

Table 3. Accuracy results

Size of dataset	Accuracy (%)
50	82
100	79
125	77
Mean	79

7 Conclusions

In this paper a theoretical prescriptive model for optimising downtime management is proposed that was derived from a systematic literature review of FSM, IoT and IQ theory. The use of intelligent algorithms and data accessibility are features of the model that can aid in the reduction of downtime. The model also supports geographically

dispersed devices and clients. From a practical viewpoint, an organisation in the smart lighting industry was used to test the model as a proof of concept. In the case of the smart lighting scenario, prior to the intervention of our study, an SMS/Ajax polling system was used that was slow and expensive due to the data costs. As a result insufficient data was provided to assist with detecting and diagnosing problems. The solution lacked real-time information and field service technicians had to rely on human ‘diagnostics’ and sometimes travelling to the smart lights in order to physically detect problems. The proposed IoT model for downtime management was used to design an architecture and to develop and implement a system prototype for optimising downtime management in the smart lighting environment.

The evaluations of the prototype revealed that web-sockets are more efficient and cost-effective than other web-based data transfer protocols such as Ajax. The implementation of a web-socket based protocol provided a low-cost data communication protocol with real-time bi-directional capabilities and fully duplex communication between a smart light and a remote server. The use of IoT-enabled communication protocols reduced the latency time and data exchange costs. Furthermore, the web-socket server implements an expert system mechanism using intelligent algorithms for data analysis. The intelligent algorithm, a C4.5 decision tree, automates fault detection and provides an issue report. The intelligent algorithms can assist service technicians to identify and diagnose problems. The practical contributions of this research are therefore the model, which can be used by FSM organisations in the implementation of IoT. The results of the evaluations revealed that the implementation of the various techniques and features of the model optimised downtime within the smart lighting environment. A problem encountered during the study related to restrictions on GSM protocols by the mobile service providers, some of which do not support the use of web-socket connections. Another challenge was inventor patents on the smart lights in the case study that restricted testing of the prototype in its natural environment. As a result only historical data was used for testing. A further limitation was that not all elements of the model could be tested due to time and resource constraints. However, the findings of this study can still be used by other researchers as a valuable source of reference when conducting similar research. The lessons learnt can be useful to other researchers and practitioners working in FSM and similar industries that can benefit from IoT.

The combination of advanced big data analytics, cloud-computing and IoT enables users to not only gather vast amounts of data but also enable them to process it without having to acquire high infrastructural costs. This leads to several opportunities for researchers in these fields. Future research directions could extend the study to include functionality such as predictive maintenance. AI mechanisms can be implemented in the model to support the prediction of faults before they occur. Additional intelligence can be achieved by interacting with other systems in the same environment that have a direct impact on the equipment’s performance. The addition of predictive mechanisms as well as enabling object interaction with other systems will transform regular equipment into a self-aware and self-learning machines, and consequently improves overall performance and maintenance management. The model serves as a reference model for standards and protocols in an IoT-based implementation in the field of downtime management within the after-sales industry. Although the study was limited

to evaluating the prototype in only one environment it provided valuable lessons that could be used by other practitioners and researchers to guide the implementation of IoT in FSM.

Acknowledgements. The financial assistance of the National Research Foundation (NRF) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at are those of the author and are not necessarily to be attributed to the NRF.

References

1. Lehtonen, O.: Taking advantage of after-sales product information in a multi-company environment. Masters thesis, Helsinki University of Technology, Department of Industrial Engineering and Management, Finland (2005)
2. Agnihotri, S., Sivasubramaniam, N., Simmons, D.: Leveraging technology to improve field service. *Int. J. Serv. Ind. Manag.* **13**(1), 47–68 (2002). <https://doi.org/10.1108/09564230210421155>
3. Petrakis, I., Hass, C., Bichler, M.: On the impact of real-time information on field service scheduling. *Decis. Support Syst.* **53**(2), 282–293 (2012). <https://doi.org/10.1016/j.dss.2012.01.013>
4. Castro, M., Jara, A.J., Skarmeta, A.F.G.: Smart lighting solutions for smart cities. In: Proceedings - 27th International Conference on Advanced Information Networking and Applications Workshops, WAINA, pp. 1374–1379 (2013)
5. Basu, C., et al.: Sensor-based predictive modeling for smart lighting in grid-integrated buildings. *IEEE Sens. J.* **14**(12), 4216–4229 (2014). <https://doi.org/10.1109/JSEN.2014.2352331>
6. Koh, L.H., Tan, Y.K., Wang, Z.Z., Tseng, K.J.: An energy-efficient low voltage DC grid powered smart LED lighting system. In: IECON Proceedings (Industrial Electronics Conference), pp. 2883–2888 (2011)
7. Kortuem, G., Kawsar, F., Fitton, D., Sundramoorthy, V.: Smart objects as building blocks for the Internet of things. *IEEE Internet Comput.* **14**(1), 44–51 (2010). <https://doi.org/10.1109/MIC.2009.143>
8. Jose, G.J., Kumanan, S., Venkatesan, S.P.: Optimize field service management through analytics. In: Proceedings of the International Conference on Advances in Production and Industrial Engineering 2015, pp. 529–534 (2015)
9. Gartner. Magic Quadrant for Field Service Management (2017). <https://www.gartner.com/doc/3808464/magic-quadrant-field-service-management>
10. Knotts, R.M.H.: Civil aircraft maintenance and support. *J. Qual. Maint. Eng.* **5**(4), 335–348 (1999). <https://doi.org/10.1108/13552519910298091>
11. Nonaka, I.: A dynamic theory of organizational knowledge creation. *Organ. Sci.* **5**, 14–37 (1994)
12. Delone, W.H., Mclean, E.R.: The DeLone and McLean model of information systems success: a ten year update. *J. Manage. Inf. Syst.* **19**, 9–30 (2003)
13. DeLone, W.H., McLean, E.R.: Information systems success: the quest for the dependent variable. *Inf. Syst. Res.* **3**(1), 60–95 (1992)
14. Schneider, J., et al.: Asset management techniques. *Int. J. Electr. Power Energy Syst.* **28**(9 SPEC. ISS), 643–654 (2006). <https://doi.org/10.1016/j.ijepes.2006.03.007>
15. Przydatek, B., Song, D., Perrig, A.: SIA: secure information aggregation in sensor networks. In *ACM SenSys*, pp. 255–265 (2003)

16. Bi, Z., Da Xu, L., Wang, C.: Internet of things for enterprise systems of modern manufacturing. *IEEE Trans. Industr. Inf.* **10**(2), 1537–1546 (2014)
17. Coetzee, L., Eksteen, J.: The Internet of Things – Promise for the Future ? An Introduction. In: *IST-Africa 2011 Conference Proceedings*, pp. 1–9 (2011)
18. Borgia, E.: The internet of things vision: key features, applications and open issues. *Comput. Commun.* **54**, 1–31 (2014)
19. Miorandi, D., Sicari, S., De Pellegrini, F., Chlamtac, I.: Internet of things: vision, applications and research challenges. *Ad Hoc Netw.* **10**(7), 1497–1516 (2012)
20. Hwang, Y.M., Kim, M.G., Rho, J.J.: Understanding internet of things (IoT) diffusion: focusing on value configuration of RFID and sensors in business cases (2008–2012). *Inf. Dev.* **32**(4), 969–985 (2016)
21. Hevner, A.R., Gregor, S.: Positioning and presenting design science research for maximum impact. *MIS Q.* **37**(2), 337–355 (2013)
22. Venable, J., Pries-Heje, J., Baskerville, R.: FEDS: a framework for evaluation in design science research. *Eur. J. Inf. Syst.* **25**, 77 (2016). <https://doi.org/10.1057/ejis.2014.36>
23. Fette, I.: The WebSocket protocol. *Internet Eng. Task Force, Request for Comments* **53**(9), 1–79 (2011). <https://doi.org/10.1017/CBO9781107415324.004>
24. Lubbers, P., Greco, F.: HTML5 web sockets: A quantum leap in scalability for the web. *SOA World Magazine, Article* (2010)
25. Da Xu, L., He, W., Li, S.: Internet of things in industries: a survey. *IEEE Trans. Industr. Inf.* **10**(4), 2233–2243 (2014)
26. Datta, P., Bhisham, S.A.: Survey on IoT architectures, protocols, security and smart city based applications. In: *8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–5. IEEE (2017)
27. Yaqoob, I., et al.: Internet of things architecture: recent advances, taxonomy, requirements and open challenges. *IEEE Wirel. Commun.* **24**(3), 10–16 (2017). <https://doi.org/10.1109/MWC.2017.1600421>
28. Lloret, J., Tomas, J., Canovas, A., Parra, L.: An integrated IoT architecture for smart metering. *IEEE Commun. Mag.* **54**(12), 50–57 (2016)
29. Kohavi, R., Quinlan, J.R.: Data mining tasks and methods: Classification: decision-tree discovery. *Handbook of data mining and knowledge discovery*. 267–276. Oxford University Press, Inc. New York, NY. USA (2002)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter’s Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter’s Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

