Design and Implementation of an Integrated Web Application for the Motor Traffic and Transport Directorate of the Ghana Police Service

Emmanuel Opoku Debrah[1], Jamal-Deen Abdulai[1], Isaac Wiafe[1] and Ferdinand Apietu Katsriku[1*]

[1]Department of Computer Science, University of Ghana
*Corresponding author: fkatsriku@ug.edu.gh

ABSTRACT

A major challenge facing most organisations is how to share data and services in a timely and cost effective manner to simplify business processes. Integrating new application modules or devices with an existing system smoothly and without any discernible errors or complications is a major issue. A framework for creating rapidly loosely coupled service applications components, the Service Oriented Architecture (SOA), which meets time and cost constraints, has been proposed. In this paper, we report on the adoption of an iterative approach to implement an integrated web based information system for the Motor Transport and Traffic Directorate (MTTD) of the Ghana Police Service. The system is developed as a set of independent web applications sharing a database to provide a single and easy point of information access. In each iteration, a composite of the system is designed and tested, thus meeting the project design objectives.

Keywords: Ghana Police Service, Service Oriented Architecture, database, web services

Background

The Motor Traffic and Transport Directorate (MTTD) is a division under the Ghana Police Service. It was formerly known as the Motor Traffic and Transport Unit. The directorate is responsible for all road safety in the Country (Motor Transport and Traffic Directorate). The Directorate became part of the Ghana police Service in 1952 and is headed by the Commander of MTTD who reports to the Inspector General of Police (IGP).

The main duties of the MTTD are to educate the public about accident precautions, train personnel to arrest drivers who violate the traffic law and regulations, and work with other stakeholders such as the National Road Safety Commission (NRSC) and Driver Vehicle and Licensing Authority (DVLA) in ensuring deriver safety on the roads. They are also responsible for the compilation and publication of all road accident statistics in the country, as well as traffic control and management.

The activities of the MTTD are classified into four categories in terms of its operations, ticket processing, insurance records, accident records, and driver licensing units. Current methods used by the MTTD are outdated, slow and provide very little possibility of integration, causing duplication of data throughout their processes (Alonso, 2004). In this work, we demonstrate how the agile development module principles may be used together with web service as underlying implementation technology to develop and integrate a system for the MTTD.

Web based information systems: The information system of the MTTD may be viewed as consisting of individuals, computer devices and equipment with the relevant software programs, a dynamic database, and organisational practices that interact in a recommended systems configuration. The system is designed to record, store, update, and expedite the automation of

data usage on a persistent basis. The data thus stored and its administration are interconnected to members of the organisation, police officers, clients (in this case the drivers), external users (of which the insurance companies are example) and other law enforcement agencies who depend on the MTTD information systems for information. The main purpose of the information management system is to meet operational needs of the unit and to facilitate information reporting and data analytics relevant to promoting road safety. The information system also permits the exchange of information among the various stakeholders (Whisenand, 1971).

Service Oriented Architecture (SOA)

Whilst there is no universal agreement in the literature on the meaning of Service Oriented Architecture (SOA), it is generally acknowledged that the underpinning notion is service-centric distributed computing. A service may be viewed as a self-contained logical representation of a business activity with a specified outcome. Such an activity must be repeatable and may be made up of other sub services. In this paper, we adopt the definition by Organization for the Advancement of Structured Information Standards (OASIS), which is believed to be more inclusive and complete than other definitions proposed in the literature. SOA is defined by OASIS as *A paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations* (W3C, 2007). We may then infer that the term capability is central to this concept. A capability may denote a set of functionalities or techniques implemented for a specific business process or sub-process, implying that a group of capabilities may also be viewed as a service. Therefore, from the perspective of SOA, a service not only refers to a set of capabilities, but also to all related capabilities which can be grouped together to be used by diverse business processes. From the foregoing, three types of services under SOA implementation may be identified (Erl, 2008a):

- An **Entity Service:** information system entities like staff, ticket, license and reports provide the foundation for this kind of activity and place a practical limitation on the context within which these entities might be used.
- **Task Service:** this is an information system service whose functional boundary is directly related to an explicitly defined business task or process.
- **Utility Service:** this is an application service providing reusable service functions of a non-business-centric nature. Such service functionalities may include event logging, reports and exception handling to other entities and task services.

The design goals of SOA are to achieve the following, among other things: loose coupling, reusability, standardized service contract, abstraction, composability, autonomy, statelessness and discoverability of services (Endrie, 2004; Erl, 2008b). The benefits associated with SOA include its ability to leverage the use of existing assets, the ease with which it can integrate and manage complexity, the more reactive and faster time-to-market that it provides and its cost reduction and increased reusability.

The SOA framework is not without its challenges and should not be seen as a silver bullet for systems design (Fuerlicht, 2006; Erl, 2008a). The adoption of this framework increases design complexity. It also requires an agreed design standard, as this increases uniformity and compatibility throughout the diverse segments of the system. There also would be the need for governance structures to be put into place to ensure that laid down processes are adhered to.

The reported work was aimed at integrating the database systems for the Eastern Regional Motor Transport and Traffic Directorate of the Ghana Police Service. A consolidated service bus was used to simplify the interoperability of applications across platforms (Web application, Mobile Application), enhance communication within the MTTD infrastructure and create an enabling ecosystem for a new layer of abstractions to be added without the need to modify the

complete system. The three services (Figure 1b) were developed as freely coupled services and dependability attributes, namely security, embedded within them to include authentication and encryption as well as reliability. An evaluation process was conducted to show that by implementing systems as web services, MTTD is able to obtain the declared advantages of an SOA model, particularly those linked with the flexibility of web services to the underlying business environment.

The challenge confronting the MTTD is to have an information system to control their operations and provide road duty officers access to the system using a mobile application to verify drivers' information. To tackle the problem, the following two solutions were proposed:

i. First, design and implement an integrated information system for the organization, which handles all the four departments of the organisation (Fig. 1a).

ii. Implement web services with mobile application to connect to the MTTD information systems. Web-services implementation was suggested as an option which facilitates the amalgamation of mobile applications with other heterogeneous web applications. Figure 1b shows the proposed integrated system.

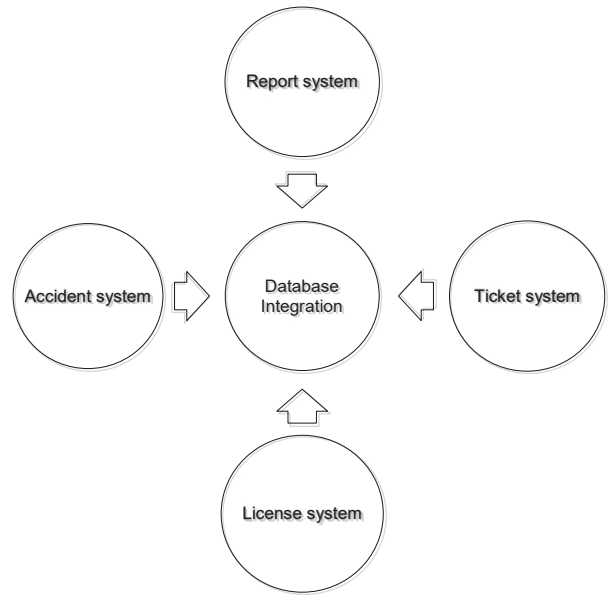

Fig. 1a: Proposed information system architecture for MTTD

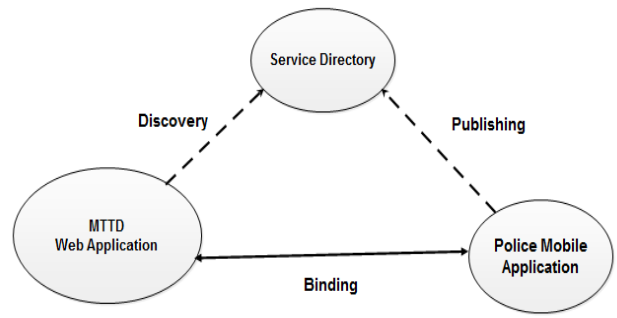

Fig. 1b: Proposed Integrated Systems

Table 1: Non-functional Requirements

SN	Non-functional Requirements	Relevance	Description
1	System Performance	Key	A key requirement is system performance; this should be adequate and fast to provide a good user experience.
2	User Friendly Environment	Key	A simple and user friendly system interface is an essential requirement.
3	Application Maintenance	Key	Clear documentation on the application and its maintenance needs to be provided.
4	Paging	Required	When dealing with large chunks of data, system should be capable of splitting these data using data paging and present the information in a user-friendly format.
5	Application Scalability	Required	System should be easily scalable; such flexibility will enable future growth.
6	Platform Independence	Required	System should be capable of being deployed on any platform.

Non-Functional Requirement

The Non-functional requirement listed in Table 1 applies to all the four systems that were designed.

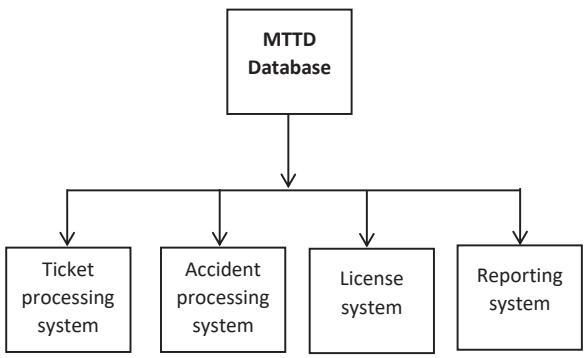

Fig. 2: Context diagram for the MTTD system

Figure 2 shows the context diagram for the entire MTTD system. These systems were designed to be hosted on the same webserver to share a common database.

Even though each system operates separately from the other, they are integrated in a way to share the same database. This means that any of the systems can run from a different webserver; once the database integration connection points to the database, it should be able to work perfectly without any problem. Any of the above systems can run independently with or without the other systems as long as the database is up and running, even though some of them depend on each other for data to work with. As an example, the license system must issue drivers' licenses and capture drivers' details before the ticket processing system can record a ticket for a driver. This doesn't mean that the ticketing processing system cannot run without the license system; it will, but it will not get the data needed to function effectively.

Again, the reporting system depends on the ticketing processing system, the accident processing system and the license system to generate the needed report for third parties including insurance companies. The reporting system can run even if all the three systems are down, but will not be consistent with Schmidt's (2005) claim. In this case, even though the system is available, it cannot be reliable (Schmidt, 2005). The entire system is managed by one administrator who assigns roles and permissions.

System Implementation and Testing

The implementation phase involved modelling the solution, coding the solution using PHP script language and MySQL database for the various web applications within the MTTD information system, and adopting a REST architectural style for building web services. The MTTD web based integrated system is implemented as several services interacting with each other in a loosely coupled fashion to achieve the intended objectives. The service is implemented on a web server which provides access to the service using HTTP protocol. Figure 3 depicts the overall system architecture.

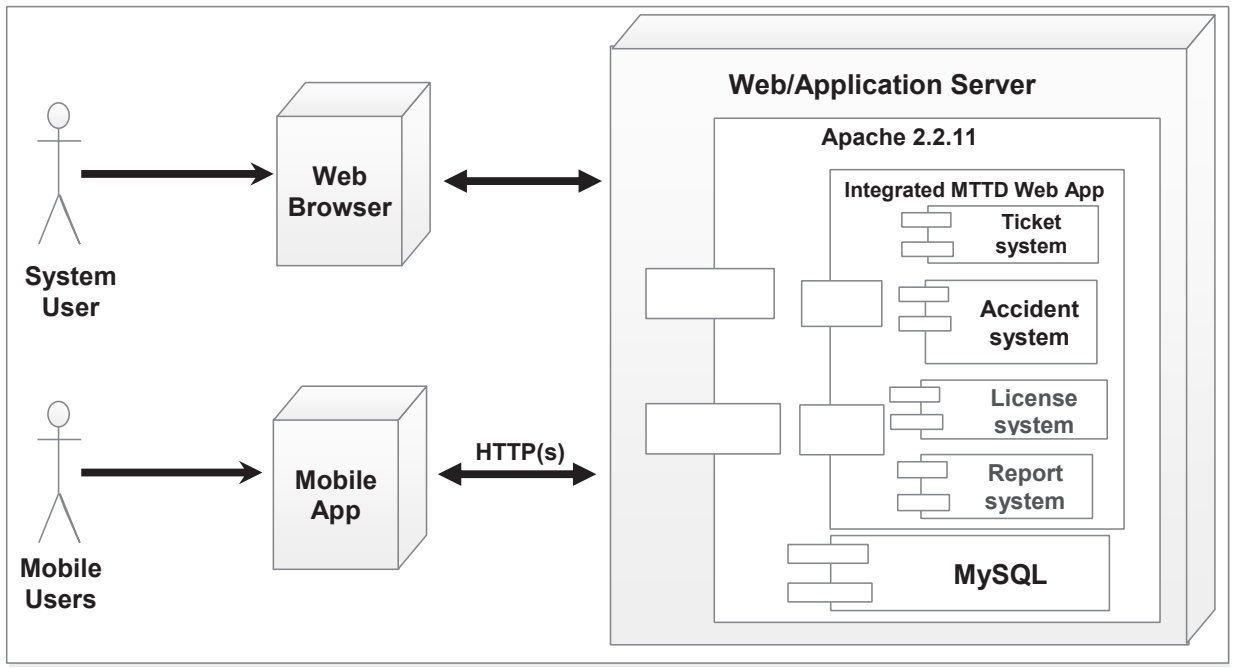

Fig. 3: Overall system architecture

The web applications were developed using PHP 5.3.0 which is a common server scripting language that offers developers the capability to develop web applications based on the Model, View and Controller (MVC) paradigm. The MVC paradigm offers a development environment that tends to disconnect the application logic "controller" from User-interface "view" and also from the data manipulation layer "model". This separation presents developers with the flexibility to alter any of the layers "controller, view or model" without requiring them to make changes to other layers. For complex systems this flexibility has great advantages. It also gives the developer a rich and varied set of scripts that helps to generate a number of common functions automatically. On the other hand, mobile application WS was developed using PHP Mysql with RestFul Web Service.

Database implementation

Storing and recalling data is one of the key elements within the integrated MTTD web applications. The MTTD web service URI acts as a resource which must contain information from the database. This information will be stored within a MySQL database, located on the same server that hosts the MTTD Information system. MySQL is an open source Relational Database Management System (RDBMS) and this informed its choice. In addition to the fact that it is freely available, it is also relatively light compared to other RDBMS such as ORACLE. The MTTD application database will consist of a number of tables to manage the ticket, accident, license and report processing.

Web Applications implementation details

As shown in Figure 2, there are four (4) main components of the MTTD web applications developed: the ticket processing system, license processing system, accident processing system and report system.

First Iteration: Implementation of Ticket processing system

At the first iteration, we implement the ticket processing system. The ticket processing system has three (3) user permission levels: the police clerk, the court clerk and the administrator. The administrator is responsible for managing user accounts and permission.

Second Iteration: Implementation of Accident processing system

At the second iteration the accident processing system is implemented. The accident processing system is the first independent web application that was integrated into the ticket processing system to share the same database. Figure 4 shows a screenshot of the database connection string that was used for the integration.

```
// Database connection info
var $Host = '127.0.0.1';
var $Port =       ;
var $Username = 'accident';
var $Password  = '@accident';
var $DbName = 'spbase';
```

Fig. 4: Database connection for accident processing system

Third Iteration: Implementation of License processing system

At the third iteration we implement the license processing system. The license processing system is the second independent web application that was integrated into the existing integrated system, which comprises the ticket processing system and accident processing system, to share the same database. The database integration was done using a PHP database connection string similar to that used for the accident processing system.

Implementation of Report system

The fourth and final iteration was the implementation of the reporting system. The report processing system is the third independent web application that was integrated into the existing integrated system to share the same database.

Implementation details of WS

To implement web services, a WS stack and APIs are required. The WS used in this work was created using RestFul web service with PHP. REST is a simple stateless architecture that generally runs over HTTP. The REST web services system produces a status code response in JSON or XML format. The main purpose for implementing the web service is to authenticate and search for drivers' license information outside the domain of the MTTD information system.

MTTD Mobile Application

In order to implement and test the web service, a small mobile application was developed to make a system call to the MTTD web services application developed in this project. This mobile application was developed on Android platform and allows an officer to check the validity of driver license information.

Testing

Driver information and officer details were verified during the tests conducted. Google chrome Advance REST client was used to test the WS created. Login request and response web services were tested and Figure 5 shows the output test result. It may be seen that the details were successfully verified within 365ms.

200 Success Loading time 365 ms

CSP: active
Origin: chrome-extension://hgmloofddffdnphfgcellkdfbfbjeloo
User-Agent: Mozilla/5.0(Windows NT 6.3; WOW64)AppleWebKit/537.36 (KHTML, like Gecko) Chrome/43.0.2357.134 Safari/537.36
Content-Type: multipart/form-data; boundary=ARCFormBoundarycqdapu96zb0529
Accept: */*
Accept-Encoding: gzip, deflate
Accept-Language: en-US, en;q=0.8
Cookie: PHPSESSID=oi32m455915bomjbfgoh8mbip5

Date: Mon, 20 July 2015 12:05:35 GMT
Server: Apache/2.2.11 (Win32) PHP/5.3.0
X-Powered-By: PHP/5.3.0
Content-Length: 50
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/json

Fig. 5: Test result of login to the web service and response time.

200 Success Loading time 324 ms

CSP: active
Origin: chrome-extension://hgmloofddffdnphfgcellkdfbfbjeloo
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64)AppleWebKit/537.36 (KHTML, like Gecko) Chrome/43.0.2357.134 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Accept: */*
Accept-Encoding: gzip, deflate
Accept-Language: en-US, en;q=0.8
Cookie: PHPSESSID=oi32m455915bomjbfgoh8mbip5

Date: Mon, 20 July 2015 12:10:07 GMT
Server: Apache/2.2.11 (Win32) PHP/5.3.0
X-Powered-By: PHP/5.3.0
Content-Length: 50
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/json

Fig. 6: Test result of search for license details web service and response time

A driver license search using web service was also conducted and Figure 6 shows the output results. As can be seen, the response times are reasonably good. By using an online free open source testing tool called Agilload (Optimise your load and performance testing costs, 2015), all services and their functionalities were tested and shown to be working. It can be seen that a driver license search was successfully conducted in 324ms.

Figure 7 depicts the output of the web service from the database in JSON format to be consumed by the mobile application and any other future application which will need driver information from the MTTD information system for implementation. A free online tool, Chrome Advance REST client, was used for this testing and evaluation.

```
        dateIssue: ''2015-06-16''
        drvName: "SAMUEL APPIAH MENSAH"
        drvDOB: "1993-04-20"
        drvAddress: "OBOC WEWEW MMER"
        drvRestriction: "Active"
}
-3: {
        drvLicenseNo: "LIGH10202"
        dateIssue: ''2015-05-12''
        drvName: "EMMANUEL APPIAH"
        drvDOB: "1989-02-07"
        drvAddress: "BOX 98111"
        drvRestriction: "Active"
}
- 4: {
        drvLicenseNo: "LO23124"
        dateIssue: ''2015-06-26''
        drvName: "MARTIN OFFEI"
        drvDOB: "1978-06-14"
        drvAddress: "KOFORIDUA"
        drvRestriction: "Active"
}
- 5: {
        drvLicenseNo: "LO23126"
        dateIssue: ''2015-06-27''
        drvName: "TIE LINCOLN"
        drvDOB: "1973-12-12"
        drvAddress: "P. O. BOX 88232
        KOFORIDUA"
        drvRestriction: "Active"
}
- 6: {
        drvLicenseNo: "6"
        dateIssue: '0000-00-00''
        drvName: ''2015-06-27''
        drvDOB: "0000-00-00"
        drvAddress: "1973-12-12"
        drvRestriction: "P. O. BOX 88232
        KOFORIDUA"
}
```


Fig. 7: JSON Output of the implemented web service

Evaluation

A number of key parameters were identified at the start for the purposes of evaluation.

Secure Login: A secure login was implemented successfully for all the various web applications, allowing multiple user accounts to access the system. Hash md5 algorithm was used to ensure secure password storage within the database and a PHP function was put together to validate user sessions upon accessing any page within the system. The requirement for a secure login with a separate user account was implemented satisfactorily. An additional security session checking script was an added advantage to the system.

A password change in the script was implemented successfully to enable users change their passwords when necessary.

File Management: One of the key functional requirements was file management. This included the ability to view, edit and add client tickets, accident files and driver licence information with ease. All of these requirements were implemented into the system along with the addition of policy categorisation, meaning that this area of functionality was met fully. It is also worth noting that the addition of touches such as a JavaScript function to change which policy type to insert, means that the final system actually delivers more than the original requirements laid out.

Reports: The reports were implemented into the system in such a way that both an HTML format (for printing or storing) and a CSV format (for use in a spread sheet program) report could be generated. This was useful for MTTD in terms of the flexibility offered with the data the system outputted, as their requirements for the layout and format could change in the future.

Data Integrity and Security

An area of concern was the future migration of data from the new system. However, choosing to utilise a MySQL database meant that exporting the data would not be a particularly challenging task – obviously conversion of this data to fit a new systems format would require some work, but the main requirement was that the data could be accessed and moved if needed – which it can. To ensure the integrity of the systems data, validation was implemented onto user input forms. Through the use of JavaScript to immediately alert the user to any incorrectly formatted data, the system was able to remain

user friendly whilst enforcing data format rules. System testing ensured that this validation worked correctly throughout and as a result, data integrity within the database was assured.

The security of the system and data was enforced through two measures.

1. A secure user login and session authentication script ensured system access was only granted with the correct credentials.
2. SQL injection and hacking attempts were prevented through the implementation of a character stripping and validation function.

Conclusion

This paper has reported on the implementation of an integrated usable web application for the various operations of the MTTD of the police service of Ghana using the SOA approach. We reviewed the concept of Service Oriented Architecture and also key web services principles, standards and components, and have shown how they could be used in application development. Four independent web applications were implemented and successfully tested. The agile development methodology was adopted for this work. To date, an industry standard software development methodology tailored specifically for SOA is non-existent. We have thus shown that it is possible to combine an iterative process with an agile methodology to achieve development goals. SOA/ Web services remain an important and one of the most promising areas in computer science. Notwithstanding the huge effort that has been exerted in the development of the fundamental standards and technologies, there remains a gap in what web services promise and what can realistically be implemented. With the current trend towards cloud based computing and services, the SOA approach may be an important methodology to consider.

References

Alonso, G., Casati, F., Kuno, H. & Machiraju, V. (2004). *Web Services. Concepts, Architectures and Application*. Springer, Berlin.

Endrie, M., Ang J., Arsanjani, A., Chua, S., Comte, P., Krogkahl, P., Luo, M. & Newling, T. (2004). *Service Oriented Architecture and Web Services.* IBM Red Book, 2004. [E-book].

Erl, T. (2008a). *SOA Principles of Service Design*. Boston, MA: Prentice Hall.

Erl, T. (2008b). *Service-Oriented Architecture Concepts, Technology and Design*. Boston: Pearson Education Inc.

Fuerlicht, G., (2006). "Enterprise SOA: What are the benefits and challenges," presented at System Integration Conference, (pp. 1-8), Czech Republic.

Motor Transport and Traffic Directorate. Available online: http://en.wikipedia.org/wiki/Motor_Transport_and_Traffic_Directorate [Accessed: June 14, 2015].

OASIS, "*Web Services Distributed Management: Management Using Web Services (MUWS 1.0) Part 1*". 2005. Available online: http://www.oasis-open.org [Accessed: June. 05, 2015].

Optimise your load and performance testing costs. Available online: http://www.agileload.com/ [Accessed: 20th, July 2015].

Schmidt, M.T., Hutchison, B., Lambros, P. & Phippen, R.(2005). The Enterprise Service Bus: Making service-oriented architecture real. *IBM Systems Journal*, 44 (4), 781-797.

W3C, (2007). "*SOAP Version 1.2 Part 2*". 2007. Available online: http://www.w3.org [Accessed: June 15, 2015].

Whisenand, P. M. (1971). Automated Police Information Systems: An Argument for Vertical and Horizontal Integration. *Journal of Criminal Law and Criminology* 62 (3).