

---

## CASE STUDY- BRITISH GAS: APPLICATION OF SERVICE-ORIENTED ARCHITECTURE

**Supun Jeevaka Dissanayake**

University of Colombo School of Computing, Sri Lanka.

---

### ABSTRACT

Business organisations must adopt to new technologies to improve business processes due to the advancement of computer technologies. Service Oriented Architectures are widely adopted by multiple organisations to maintain their service quality and obtain a competitive edge. This research paper proposes a Service Oriented Architecture design to improve services of British Gas, which is one of the largest utility companies in the United Kingdom.

---

### INTRODUCTION

Service-Oriented Architecture (SOA) is a software engineering technique where the component structuring, development, service structuring and asset structuring is designed using core principles of message passing (Ramachandran, 2016). Moreover, it provides the ability for software systems to use discoverable and published interfaces to facilitate services developed in a network. This research paper follows the development process of a prototype application with Service-Oriented Architecture functionalities for British Gas. It initially identifies both functional and non-functional requirements of British gas through an evaluation process. Then it follows web service standards set by Erl (2005) such as identifying service functional descriptions, task analysis and service classifications to identify the service structure. Moreover, the designing criteria will be carried out by developing use-case diagrams to identify the sequence of actions, developing component diagrams to identify software structure and SoaML diagrams to identify the SOA. Furthermore, a business process model will be developed for one of the British Gas service requirements and that service will be simulated with costs and durations to evaluate its working functionality and efficiency. Finally, one of the SOA requirements will be implemented as a .NET application using Microsoft Visual Studio. Dissanayake (2017) proposed MLSOA which is a methodology to use machine learning to improve business processes via a maturity model for organisations following service-oriented architectures. Therefore, this model will be further explored and applied for British Gas service-oriented architecture to enhance its capabilities.

### CASE STUDY: British Gas

British Gas have decided to change their software systems into a service-oriented architecture (Donnelly, 2015). They believe that it will allow them to stay competitive with the current trends in technology and provide a better service to their customers. SOA would allow British Gas to enhance their IT department to provide more value-added services. British Gas divides their business application into three categories: functional services, information services and tasks requiring human decisions (Donnelly, 2015). Thus, SOA should be applied across these categories. It is extremely important to identify the security criteria of the application when the services are transferred to SOA. Moreover, British Gas is planning on transferring their web services into a cloud platform (Donnelly, 2015). Thus, it allows British Gas to provide

services to customers' error-free through multiple devices via cloud services such as SaaS. Therefore, this research paper identifies requirements for a SOA design architecture for British Gas and provides solutions to achieve these requirements through a SOA design process.

### **Service Requirements**

Service requirements identify services that British Gas intend to apply to their SOA. Thus, initially, service requirements of British Gas must be recognised. Following subsections depict functional service requirements for British Gas.

### **Customer Handling**

Customer handling is one of the main services that British Gas should provide through their SOA. Customer handling can be further subdivided into three sections to simplify its term.

- Create customer: a new user should be added to the cloud data store. This service is applicable when a customer applies for a new product. Then the customer should get a quote from British Gas. After customer and BG agree on the quote provided, the customer should be accepted by the organisation.
- Update customer: the customer should be able to update their personal information through the system. For example, the customer should be able to update their address if they moved into a new property.
- Delete customer: there should be a functionality to remove customer information from the database. For example, if a customer decides to leave the supplier, there should be a process to finalise all the bills and close the account.

### **Bill payments**

Bill payments are essential for a utility service such as British Gas. It can be further subdivided into three sections,

- Receive bill: The process of customer receiving the bill from British Gas (attributes: date, meter reading, cost, usage, house number, postcode)
- Submit bill: The process of customer paying the bill to British Gas (attributes: date, meter reading, house number, postcode)
- Update bill: The process of customer updating any arrears payments to British Gas (attributes: date, meter reading, house number, postcode)

### **Utility Catalogue**

The customer should be able to see information about utility services that British Gas provide and understand these services in depth.

### **Make a complaint**

The process of allowing customers to submit a complaint through the website. This could be further subdivided into two sections,

- Written Complaint: the customer should be able to submit a written complaint through the website and receive a written reply to it via email within 2 hours.
- Online Chat: the customer should be able to discuss or make a complaint via online chat system, thus they can receive instant service by reducing the waiting time for a reply through email.

### **Technical support and Emergency Services**

Customers should be offered technical support and emergency services when necessary. This service should also provide two services similar to the complaining process.

- Written Complaint: the customer should be able to submit a written technical support query through the website and receive a written reply. However, if the situation is an emergency. The reply should come within half an hour.
- Online Chat: the customer should be able to discuss a technical problem or an emergency via online chat system, thus they can receive instant service by reducing the waiting time for a reply through email.

**Upgrades**

British Gas should notify customers any upgrades that they are entitled to obtain through the online system.

**Appointment Booking**

The customer should be able to book an appointment with an engineer through the system. Then the customer should be able to view the appointment online, reschedule the appointment and cancel the appointment.

**Browsing and Searching**

The customer should be able to browse and search all the latest facilities that would help them to enhance their experience with British Gas and reduce costs.

**Compare**

The customer should be able to compare different pricing schemes and offers available on products to reduce their current price or to improve their product experience.

**Energy usage control**

The customer should be able to check their energy usage online. This information should be fed into the online system through smart meters and smart heating systems. Then the system should automatically identify ways to improve the energy usage through machine learning and feed that information back to the customer.

**Non-functional requirements**

In addition to functional service requirements, the system should also contain non-functional requirements. They are,

Non-functional Requirement	Description
Reliability and robustness	The system must be reliable and robust without any breakdowns. It should provide services to the customer without any delay.
Security	The system must be very secure since it contains confidential information about customers. The back-end system of the application should be only accessible for authorised personnel of the organisation. Every customer account should be secured with advanced password encryption mechanisms and a wide variety of security tools to encrypt personal information in the cloud server.
Availability	The system must be accessible to the user 24/7

Flexibility/usability and efficiency	The system should be easy to use. For example, a customer who lacks in technical knowledge should be able to easily understand how to use the system.
Reusability	The system should be developed by using CBSE (Component Based Software Engineering) methodologies

**Table 1: Non-Functional Requirements**

### Service Functional descriptions

Once the services are identified, the next stage is to develop a service interface (Sommerville, 2010). Therefore, service operations and their parameters must be identified. Furthermore, it is important to minimise the number of message exchanges between services to improve the service quality (Ankolekar, 2002). Table 2 depicts the list of operations and their descriptions.

Operation	Description
“Create”	This operation allows the customer to create a new account and signs up to become a new British Gas customer. This information will be passed on the relevant personnel in BG and the customer request will be assessed and accepted if they meet requirements. Then the customer will be told to choose a quote for the supply.
“Edit”	This operation allows the customer to update personal information.
“Delete”	This information allows the customer to request an account cancellation process. Then the customer information will be assessed by BG and check for all the acceptable cancellation criteria (e.g.: no arrears, bills up to date etc.). Once all the cancellation criteria are met, the account will be cancelled.
“Receive Bill”	This operation returns the bill to the customer. The customer will have precise information about how the bill was formed.
“Submit Bill”	This operation allows the customer to make an online payment through the online system.
“Update Bill”	This operation allows the customer to make an online payment to cover any arrears that are due on their bill.
“Catalogue Access”	This operation allows the customer to assess a list of products that BG offer. The customer should be able to download any desirable list in a pdf format.
“Make Complaint”	This operation allows the customer to make a complaint about the service. The customer must provide the customer id and complaint reasoning. This information will be passed onto relevant British Gas along with the date and time of the complaint. A two-hour deadline will be set and BG must reply with a valid answer.

“Tech Support”/ “Emergency Service”	This operation allows the customer seek help from a technical personal in British Gas. The customer must provide the customer id and a description of the issue. This information will be passed onto British Gas along with the date and time of the complaint. A half an hour deadline will be set and BG personal must reply with a valid answer.
“Upgrade”	This operation returns any service upgrades that the customer should be able to obtain from British Gas.
“Appointments”	This operation allows the customer to make an appointment with a British Gas Engineer. The customer should provide attributes such as issue, customer id, appointment date, appointment time etc. Moreover, the customer can reschedule and cancel the appointment.
“Search”	This operation searches any query that the customer wants to find out in the website and returns a list of items that matches the searched expression.
“Compare”	This operation provides comparisons between different quotes offered by British Gas in terms of Electricity and Gas Supply.
“Energy Control”	This operation returns data about energy efficiency in the household and provide guidance on which areas to improve.

**Table 2: Service Functional Descriptions**

After service description identification, service parameters should be defined (Sommerville, 2010). Following table depicts the parameters for creating, update and delete services for British Gas Customers.

Operation	Input	Output	Exceptions
Create Customer	Creation of, Id First name Surname Age Email Address	URL of the web page with Success Message	Invalid first name, surname, age, email or address
Update Customer	Update, Id First name Surname Age Email Address	URL of the web page with Success Message	Invalid first name, surname, age, email or address
Delete Customer	Removal of, Id First name	URL of the web page with Success Message	Failure to remove customer due to error in identifying

	Surname Age Email Address		customer id
--	------------------------------------	--	-------------

**Table 3: Input and Output messages for UML**

### Service Task Analysis and Service Classification

The basic principle of SOA is that the service could support business processes. Erl (2005) proposes a methodology to separate services into three different categories to identify service candidates that are best suited to the organisation. They are utility services, business services and coordination or process services.

- Utility services: services that provide general functionality for different business procedures (Sommerville, 2010).
- Business services: services that is associated with certain business purposes (Sommerville, 2010).
- Coordination or process services: services that are related to general business procedures that contain different activities and actors (Sommerville, 2010).

Moreover, Erl (2005) suggests that a service can be either task or entity oriented: entity-oriented services are objects and task-oriented services are generally associated with a specific activity. Therefore, table 4 is developed to differentiate the identified requirements into above categories.

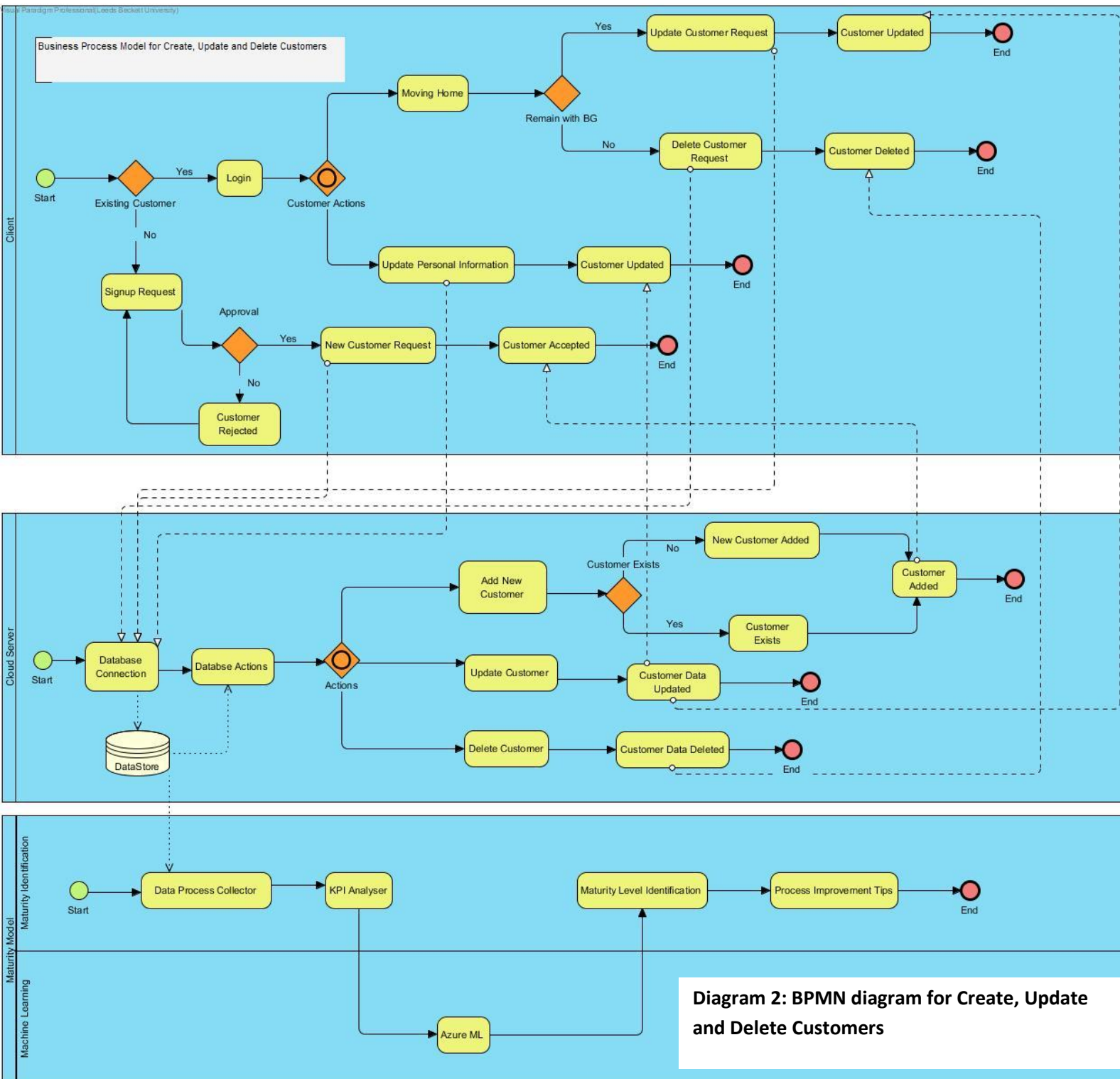
Service	Task	Entity	Utility	Business	Coordination
Create a new customer		Yes		Yes	
Update customer		Yes		Yes	
Delete customer		Yes		Yes	
Receive Bill	Yes			Yes	
Submit Bill		Yes		Yes	
Update Bill		Yes		Yes	
Catalogue Access		Yes	Yes		
Make a complaint	Yes				Yes
Tech Support Request	Yes				Yes
Emergency Service Request	Yes				Yes
Upgrade Notice		Yes		Yes	
Appointment Booking	Yes		Yes		
Search items		Yes	Yes		
Compare items	Yes		Yes		
Energy Control	Yes		Yes		

**Table 4: Service Classification**



### Business Process Model using BPMN Notation

Business process models are used to depict a set of services carried out within a business organisation (Dijkman, 2008). Thus, it allows the understanding and analysis of business processes. Furthermore, a thorough understanding of business process models in an organisation allows the increase of organisation efficiently and profitability (Dijkman, 2008). On this occasion, a single service requirement is used to depict a business process model for a singular service within British Gas, which is customer handling (creating, updating and deleting process of the customers). Following BPMN diagram is developed to depict this process by using Microsoft Visual Paradigm.



**Diagram 2: BPMN diagram for Create, Update and Delete Customers**

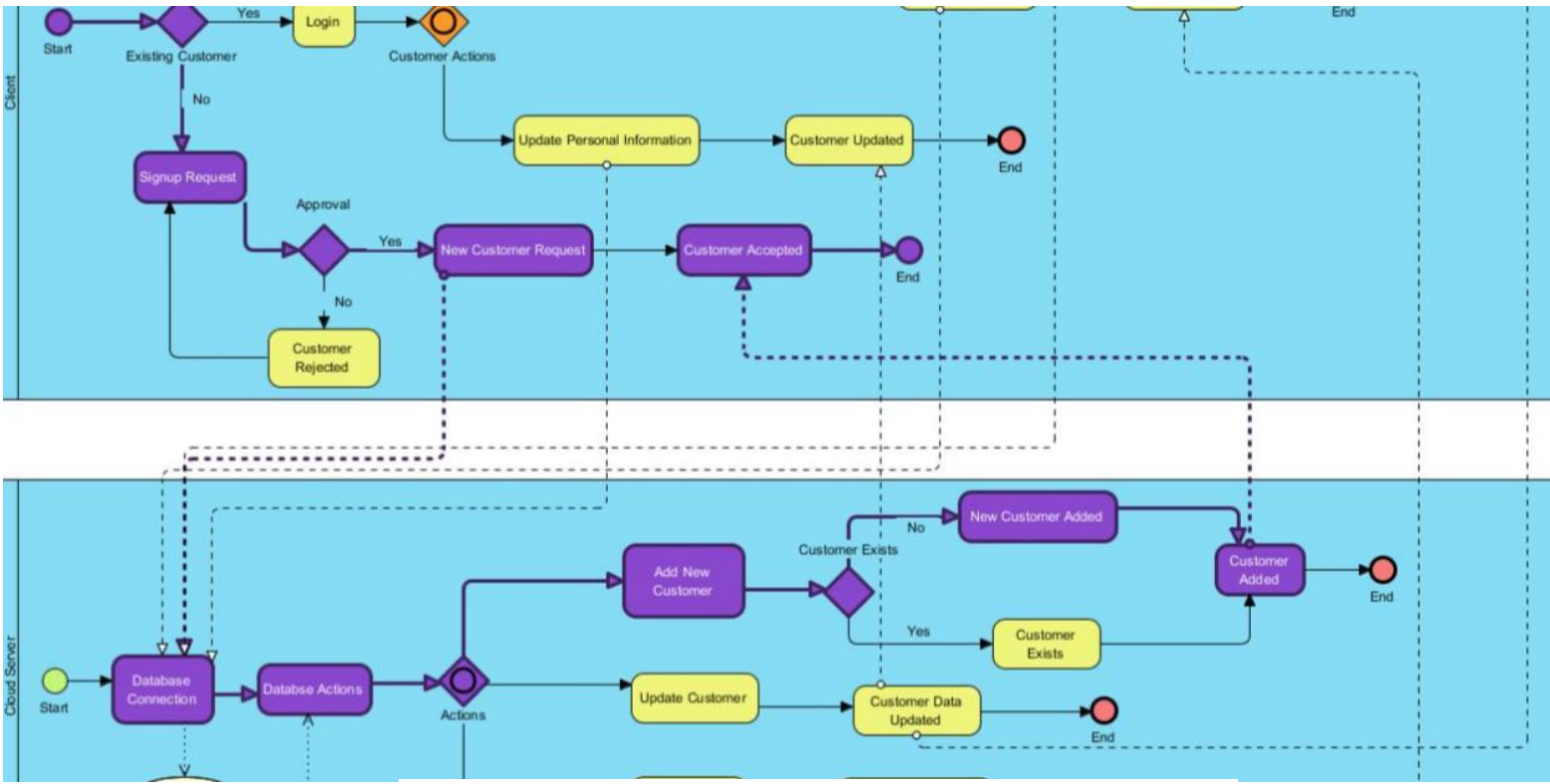
This BPMN diagram contains three separate pools: Client, Cloud Server and Maturity Model. The client pool allows the customer login or signup to the service. After the customer is logged in, it provides the functionality to either delete the account in an instance of service transfer or update the account in order to change the personal information about the customer. The application is deployed in the cloud server, which is being depicted by the second pool. Cloud Server, contains a database to store customer data, thus servers are not required to be maintained by British Gas. The cloud server depicts the business processes of data being added, updated and deleted from the system depending on the customers' requests obtained from the client side of the application.

Finally, the third pool depicts the Maturity Model of the application. This is developed as a result of maturity model specification developed by Dissanayake (2017), which allows the identification of maturity level of organisations carrying out Service Oriented Architectures using machine learning algorithms. This pool consists of two lanes: maturity identification and machine learning. As the customers use services provided by British Gas through its service-oriented architecture, data collected at the database will be formatted and transferred to the data process collector. Then this information will be passed through algorithms provided by Azure ML machine learning platform, which will allow the identification of maturity level of the organisation. Furthermore, it will provide advice in terms of areas that the organisation must improve in order to enhance the business processes of the organisation.

For example, one of the examples of the services depicted in this BPMN process model is signing up a new customer. Initially, the customer makes a sign-up request. Then approval process from the British Gas should be carried out. If the customer is approved to be a BG customer, that information will be passed to the cloud server. The cloud server will establish a connection with the cloud database. Then it will carry out a test to identify whether the user already exists. If the user exists, they will be informed and otherwise, the new user will be added to the database and the user will be notified about their acceptance. Subsequently, information about the user sign up process will be transferred from the database to the maturity model to aid the maturity level identification process, which acts as its own service entity.

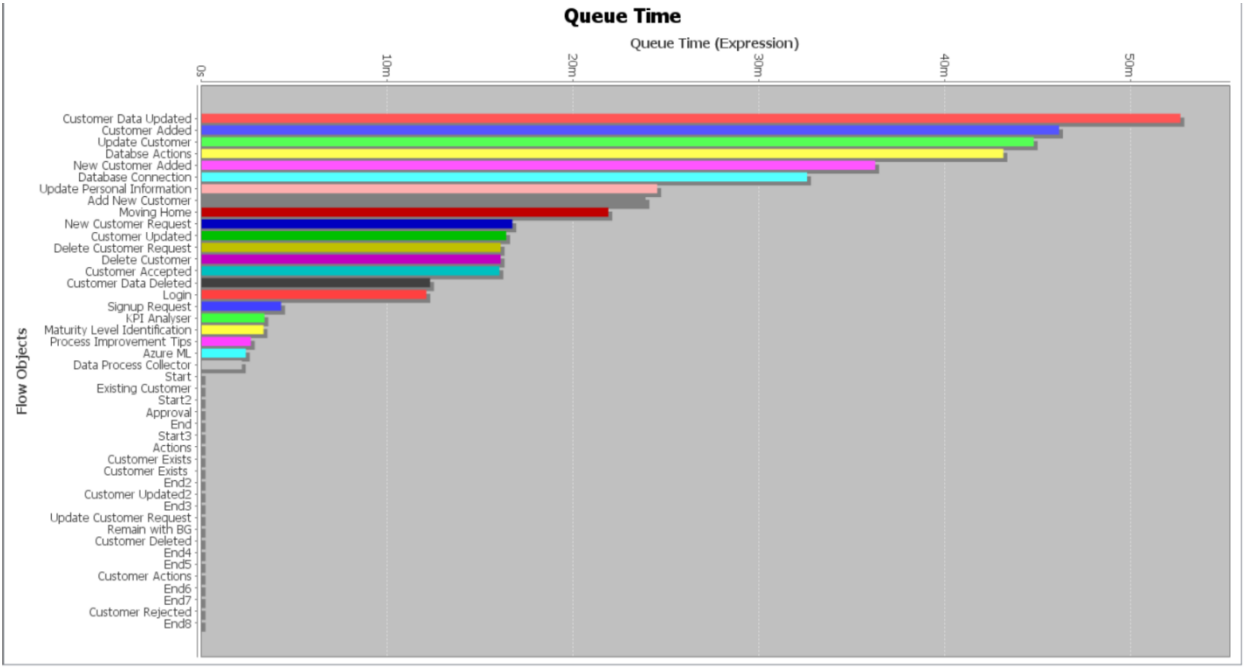
### **Business Process Simulation**

Business process simulation is the process of evaluating the BPMN model by exploring the effect of the services proposed through the business process model. It allows the identification of resource consumption such as human resources, cost, devices etc. (Tumay, 1996). Moreover, the result obtained from the business process simulation allows the organisation to rethink any of the issues that could cause via the services that they are intending to develop. On this occasion, business process simulations are carried out for creating, updating and deleting customers and maturity model identification process following the information provided through the BPMN model depicted in diagram 2. Diagram 3 depicts one example of a simulation.



**Diagram 3: Simulation for the service “Create new user”.**

Visual paradigm allows the user to assume how much money that each task will cost and how long it will take to carry out that task. Therefore, assumptions were made regarding cost and time associated with each task. The final step is to run the simulation. When the simulation is completed, it depicts the duration that it will take to carry out the tasks that were set and the amount of money that it will cost the organisation. Furthermore, it develops simulation charts to graphically depict this information. It develops charts to show completion against time, resource usage against time, queue time, cost per flow object, cost per scenario and time against cost. Following diagrams depicts each of these graphs that were developed for the



**Diagram 4: Queue Time Chart**

above BPMN model.

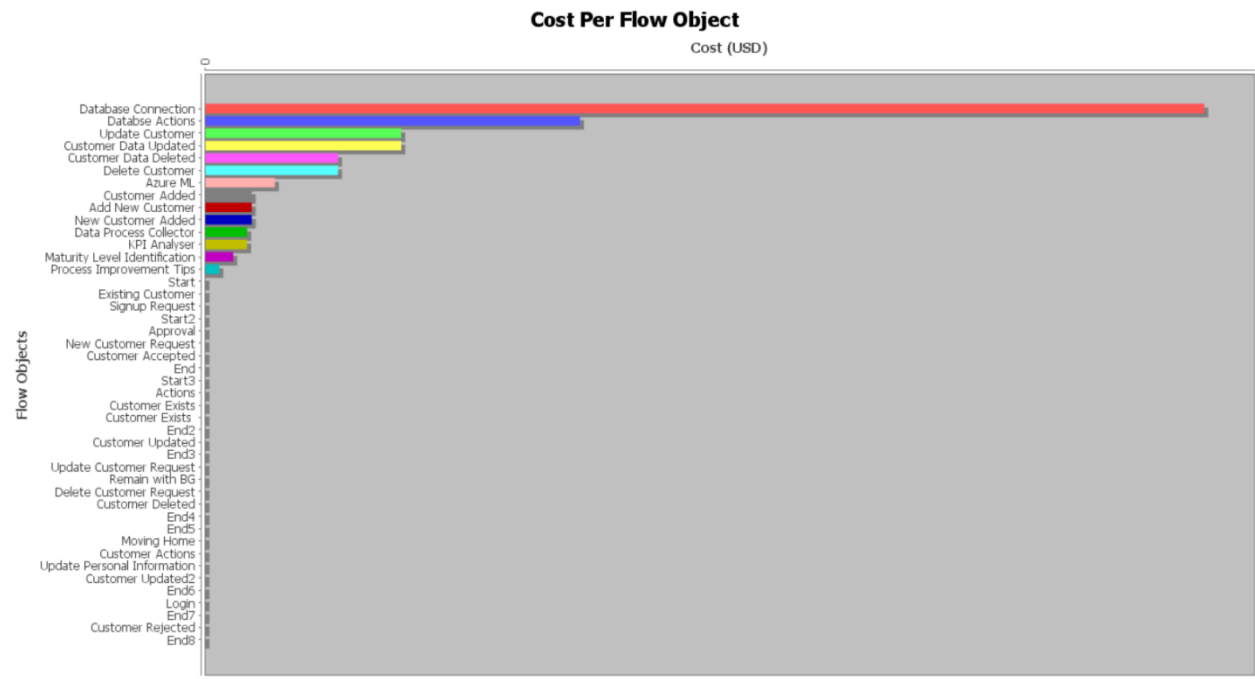


Diagram 5: Cost Per Flow Chart

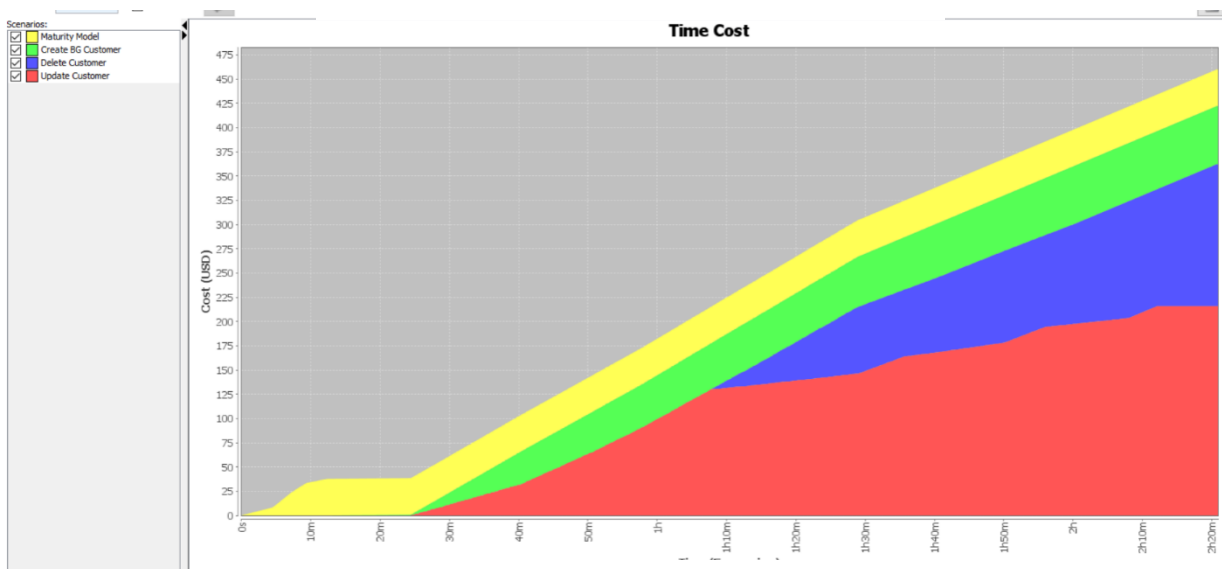


Diagram 6: Time and Cost Chart

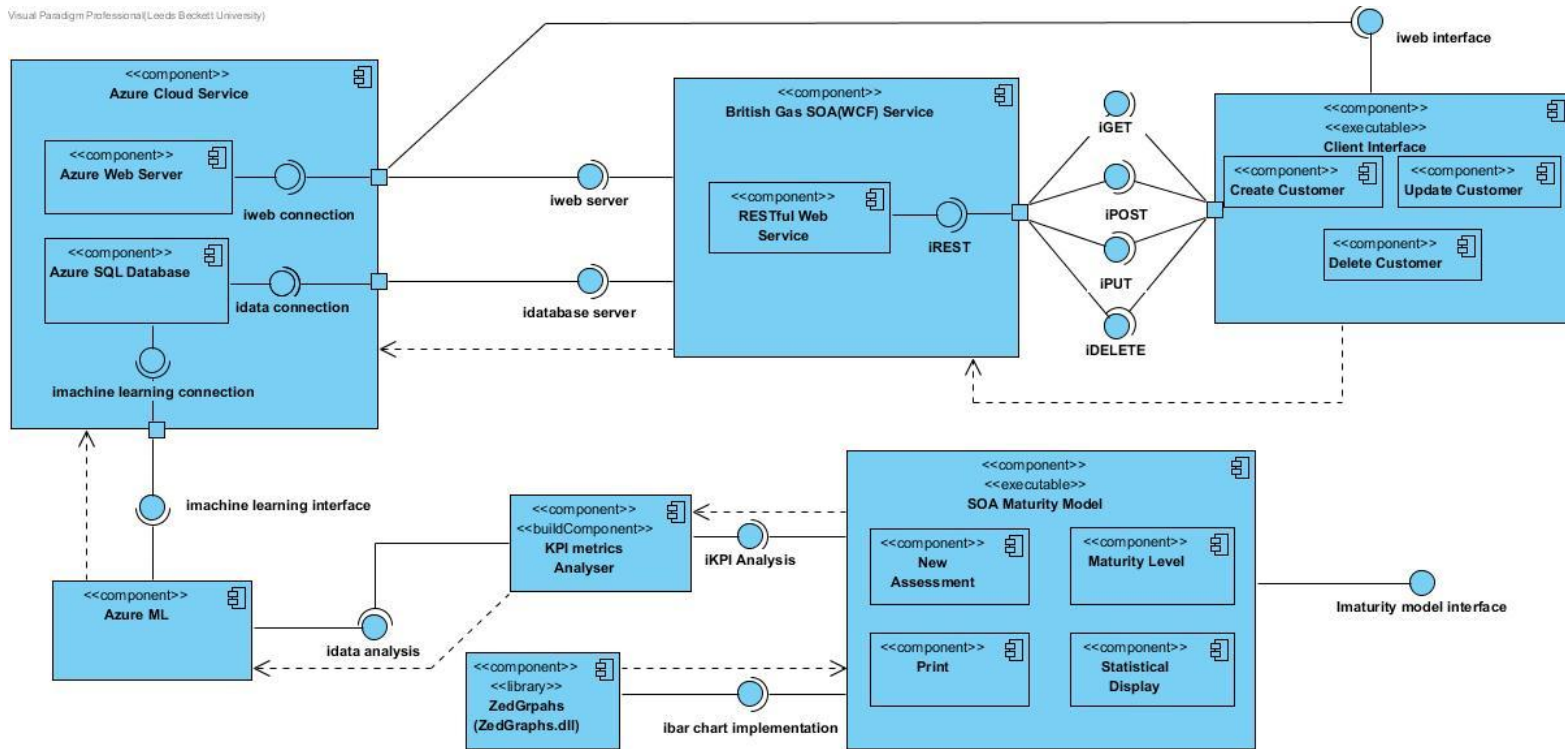
## SOA Design for British Gas

### UML component diagram for services

The component model must be recognised prior to the service development (Weinreich et al, 2001). It is important to understand how each component would function and their dependencies, thus the system architecture will be much clear. Sub-level components will depict all the internal structures of the identified components (Weinreich et al, 2001). Moreover, interface types between components should be identified to depict the manner in which the system is connected to each component (Weinreich et al, 2001). Diagram 7

implicates the component diagram developed for British Gas SOA architecture proposed through this research paper. It is being developed using Unified Modelling Language provided through Microsoft Visual Paradigm. It implicates the service architecture through 7 main components along with their internal components. The diagram depicts provided and required interfaces through the standard lollipop and socket notations (Kobrym, 2000). The usage of ports implicates the connection between internal components and main components. Furthermore, components dependencies were added to outline the way components are dependent to each other in order to function successfully.

Visual Paradigm Professional (Leeds Beckett University)



**Diagram 7: Components Diagram**

Diagram 7 depicts the client interface of the application. It will be developed as a Model View Controller through Microsoft Visual Studio using C#. It will allow British Gas Customers send requests in order to create, update and delete their user accounts. It depicts the component that allows the data to be transferred through the service. It will be developed as a Windows Communication Foundation (WCF) service application through Microsoft Visual Studio using C#. WCF framework allows the transfer of data between applications using RESTful web service (Zhang et al, 2009), which is further represented through the internal component. These data will be transferred in the form of JSON via GET, POST, PUT and DELETE protocols. It depicts the Azure cloud service component. Microsoft Azure is a cloud storage platform developed by Microsoft that allows users to build, develop and manage applications (Qian et al, 2009). It provides a wide variety of functionalities to support service oriented architectures. Azure web server will be used to deploy the client interface web service depicted in diagram 7. Moreover, data processed by the WCF service depicted in diagram 7 will be stored in Azure SQL database. Diagram 7 depicts the connection of Azure Machine Learning platform with the Azure Database. This allows data collected from the database to be passed into Azure ML. This component is used to aid the maturity level

identification of British Gas work processes by the SOA maturity model introduced by Dissanayake (2017). Azure ML allows big data analysis processes using ML algorithms. Diagram 7 depicts the KPI metrics analyser which is connected to Azure ML component. It will identify which key process indicators are achieved at each maturity level. Diagram 7 depicts the Zed Graphs components which are an open source reusable software component that draws graphs such as line graphs, pie charts, tutorial charts, bar charts and special charts to depict the graphical analysis of the data sets (Yigang, 2012). Thus, it will provide a visual implantation of the maturity level achieved by the organisation. Diagram 7 depicts the SOA maturity model component, which will process data from components depicted in diagram 7 and depict the maturity level of British Gas and areas that the organisation must improve to achieve a higher level. It will follow SOA maturity principles proposed by Dissanayake (2017).

### SoaML

SoaML allows the understanding of services and the service architecture. It is the process of developing the service architecture via Unified Modelling language (UML) standards set by Object Management Group (OMG) (Delgado et al, 2010). Moreover, SoaML allows the service to be divided into three functions: service interface, simple interface and service contract. On this occasion, SoaML is used to demonstrate the service of creating a new customer account through British Gas Service Oriented Architecture. Following subsections depict how British Gas service structure can be implied through SoaML. Specifically, they will depict one of the customer handling services identified in the service requirements section: provision of memberships to new customers. Furthermore, Thomas Erl's design techniques are considered when developing following SoaML diagrams.

### Service Interface Diagram

Service interface diagram can be used to depict service specification model and bi-directional services (Elvesæter et al, 2010). It displays the communication process between the consumer and the provider, which is one of the SOA design principles that was proposed by Erl (2005). This diagram shows the service interface for the membership obtaining criteria for British Gas. As it depicts, the customer is the membership requester and the service provider is the membership acceptor. Moreover, the outcome of this process is depicted in a separate interface: membership status.

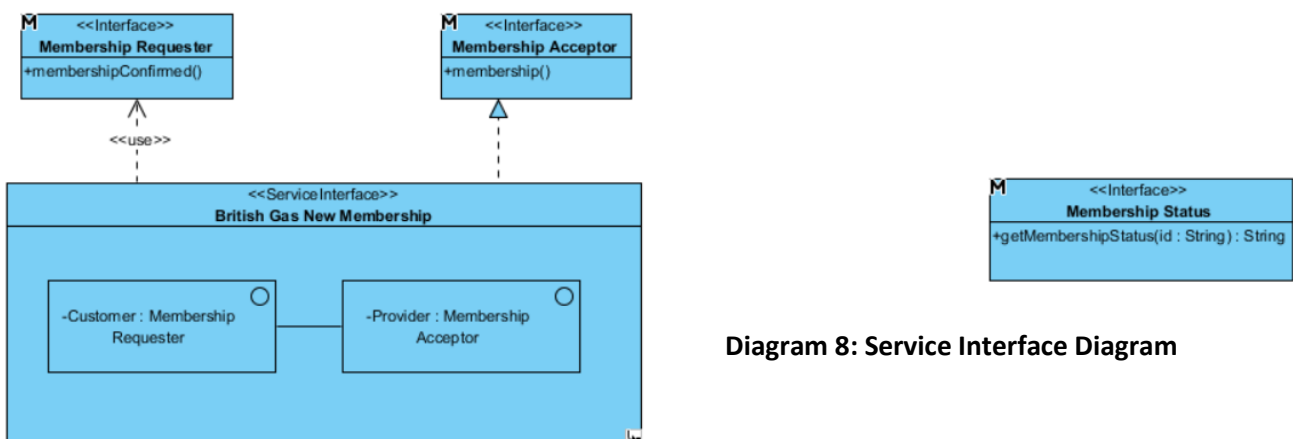


Diagram 8: Service Interface Diagram

### UML Sequence Diagram

UML sequence diagrams are used to display service choreography (Elvesæter et al, 2010). Thus, it shows the messaging process between the web service and the consumer to carry out the service requirement. This diagram implicates the sequence of action carried out between the potential customer and the service provider where the customer requests for a membership and the provider (British Gas) accepts it. It follows Erl's (2005) design principles since it depicts the message exchange process between two interfaces.

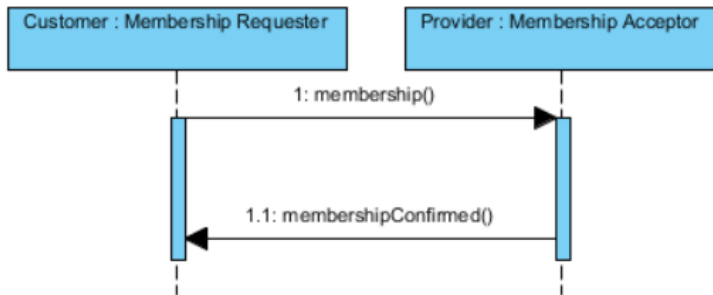


Diagram 9: UML Sequence Diagram

### Service Participant Diagram

Service Participant diagrams are used to depict components that provide and consume services (Elvesæter et al, 2010). These components are people involved in the service, company, system, components etc. (Elvesæter et al, 2010). This diagram shows two parties that are involved in the service and the how messages will be interconnected to each other.

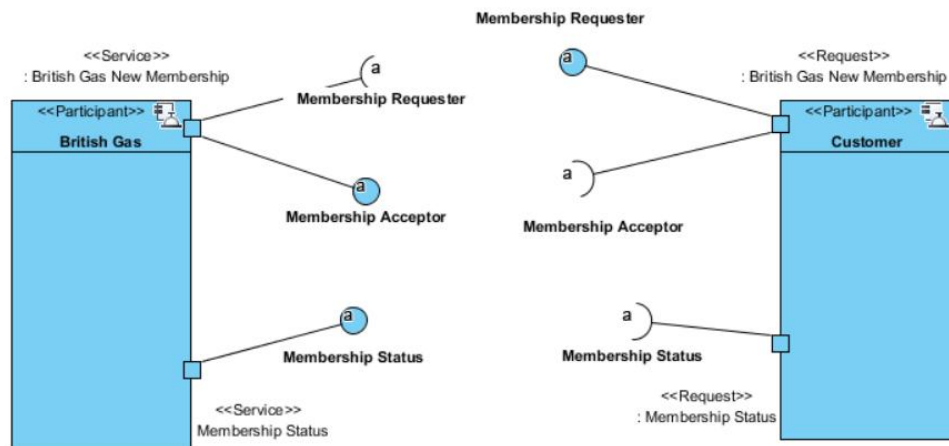
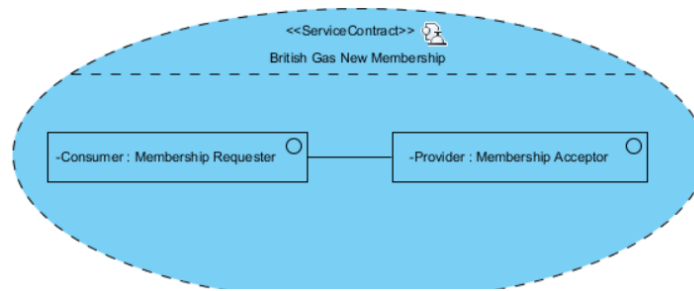


Diagram 10: Service Participants Diagram

### Service Contract Diagram

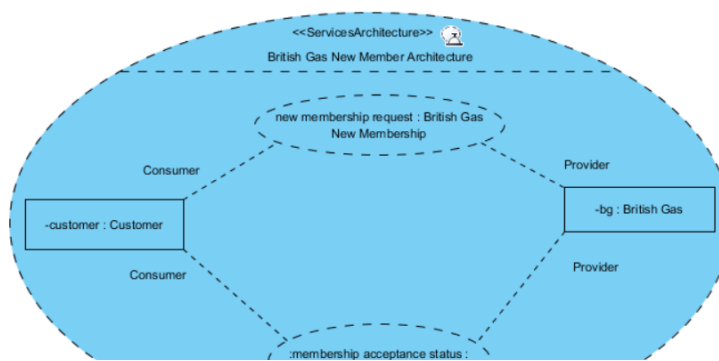
Service contract diagram is designed to depict the agreement between different interfaces and personnel associated with the service (Elvesæter et al, 2010). Moreover, all the services associated with the system must agree with each other for the service to function (Erl, 2005). This diagram implicates the agreement between British Gas and Customer when getting a new membership.



**Diagram 11: Service Contracts Diagram**

### Service Architecture Diagram

Service Architecture diagram is designed to display the way that each aspect of the service will work together to provide the required service following Erl's (2005) principles. Thus, it implicates the connections between the specified roles of the customer and British Gas that must be followed to complete the membership process.



**Diagram 12: Service Architecture Diagram**

### Service Implementation

The prototype tool implicates the criteria that were discussed through SoAML diagrams, components diagrams and BPMN models depicted in previous sections. It allows the customer to create a British Gas account, update their British Gas account and Delete their British Gas account. These processes were identified as service requirements in the service requirements section of the research paper. It is designed as a WCF (Windows Communication Foundation) service through Microsoft visual studio by using C# programming language. WCF allows the development of service-oriented applications that has functionalities to send data as asynchronous messages from one service destination to another (Zhang et al, 2009). Therefore, this ability of WCF application is applied to develop a RESTful web service that supports the user handling criteria of British Gas. Following diagrams depicts three main interfaces of the prototype, which are customer list display, customer additions, customer edits and customer deletion.

## British Gas Customers

[Add new Customer](#)

Id	Firstname	Surname	Age	Email	Address	Option
1	Supun	Dissanayake	22	supun@gmail.com	26, MonkBridge Street, Headingley, Leeds, LS6 4HL	<a href="#">Delete</a>   <a href="#">Edit</a>
2	James	Cameron	50	james@gmail.com	27, MonkBridge Street, Headingley, Leeds, LS6 4HL	<a href="#">Delete</a>   <a href="#">Edit</a>
3	Tony	Blair	55	tony@gmail.com	28, MonkBridge Street, Headingley, Leeds, LS6 4HL	<a href="#">Delete</a>   <a href="#">Edit</a>
4	Jackie	Stewart	80	jackie@gmail.com	29, MonkBridge Street, Headingley, Leeds, LS6 4HL	<a href="#">Delete</a>   <a href="#">Edit</a>
5	Bob	Marley	65	bob@gmail.com	30, MonkBridge Street, Headingley, Leeds, LS6 4HL	<a href="#">Delete</a>   <a href="#">Edit</a>
6	Lewis	Hamilton	32	lews@gmail.cm	31, MonkBridge Street, Headingley, Leeds, LS6 4HL	<a href="#">Delete</a>   <a href="#">Edit</a>
7	Jenson	Button	36	jb@gmail.com	32, Monkbridge Street, Meanwood, Leeds	<a href="#">Delete</a>   <a href="#">Edit</a>

### Add New Customer

Firstname   
 Surname   
 Age   
 Email   
 Address

### Edit Customer

Id 1  
 Firstname   
 Surname   
 Age   
 Email   
 Address

Diagram 13: Service Implementation

### WCF Service Development

Initially, a WCF service application called “British Gas Services” is created via Microsoft Visual Studio. Then a database table is developed for customer data via Microsoft SQL Server and referenced to the WCF application. It consists of 6 columns: id, first name, surname, age, email, address. A service contract is developed to provide services that will be available through the prototype. Then operation contracts were developed to get all customers, get a single customer, create a customer, edit customer and delete a customer. This is developed as a REST service and GET, PUT, POST and DELETE methods were used and JSON is used as the data interchange format. Then methods were developed to implement services.

Following this procedure, the webconfig file is modified by adding “services”, “behaviours” and “end point behaviours” to show how the service would run on the browser. After this modification, the web service is deployed. Diagram 14 depicts the JSON message retrieved for “find all” function. Moreover, diagram 15 depicts how the data can be filtered by the id by using the “find” function.

← → ↻ localhost:52373/ServiceCustomer.svc/findall

```
[{"Address": "26, MonkBridge Street, Headingley, Leeds, LS6 4HL", "Age": 22, "Email": "supun@gmail.com", "Firstname": "Supun", "4HL", "Age": 50, "Email": "james@gmail.com", "Firstname": "James", "Id": 2, "Surname": "Cameron"}, {"Address": "28, MonkBridge Street", "Id": 3, "Surname": "Blair"}, {"Address": "29, MonkBridge Street, Headingley, Leeds, LS6 4HL", "Age": 80, "Email": "jackie@gmai", "Age": 65, "Email": "bob@gmail.com", "Firstname": "Bob", "Id": 5, "Surname": "Marley"}, {"Address": "31", "Age": 32, "Email": "lews@gmail.cm", "Firstname": "Lewis", "Id": 6, "Surname": "Hamilton"}, {"Address": "32, Monkbridge Stree", "Age": 36, "Email": "jb@gmail.com", "Firstname": "Jenson", "Id": 7, "Surname": "Button"}, {"Address": "27, Monk Bridge Stree", "Age": 21, "Email": "jb@gmail.com", "Firstname": "Jack", "Id": 14, "Surname": "Daniels"}]
```

Diagram 14: JSON Result for find all function

← → ↻ localhost:52373/ServiceCustomer.svc/find/1

{"Address": "26, MonkBridge Street, Headingley, Leeds, LS6 4HL", "Age": 22, "Email": "supun@gmail.com", "Firstname": "Supun", "Id": 1, "Surname": "Dissanayake"}

**Diagram 15: JSON Result for find function**

Moreover, visual studio automatically generates the WSDL file for these implemented services. WSDL (Web Service Definition Language) is an XML-based language that implicates document-oriented services as networking services endpoints (Curbera et al, 2002). WSDL mainly defines three sections of the web service: service functionality, communication structure and the location (Curbera et al, 2002).

### **MVC Client-Side Application**

After the development of the service side of the SOA application, the client-side app was developed. This was developed in a form of a .NET web application via Microsoft Visual Studio by using C# as a programming language. Moreover, it is developed by following the Model View Controller (MVC) architecture.

### **Service example**

After developing both the SOA service functionality and the MVC Client front end, the prototype application is ready for deployment. Therefore, when the application is deployed, the first screen of the diagram 13 will appear on the web browser.

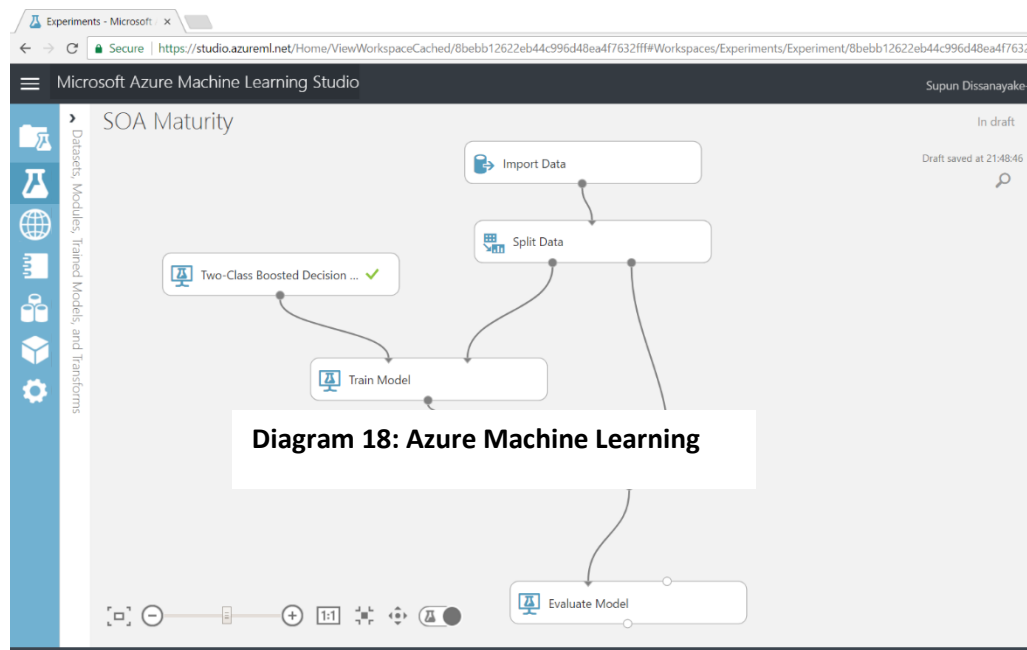
A new customer can be added by clicking the “Add new Customer” link. When this link is clicked, second screen of the diagram 13 will appear and new customer details can be added to the system. Then this new information will be displayed on the customer list. Then it can be edited by clicking the “Edit” button, which would lead the user into the third screen of the diagram 13. Moreover, the user can be deleted by clicking the delete button, which would clear all the details of the customer.

### **Microsoft Azure**

The final part of the development process is to publish the application in Microsoft Azure. Therefore, initially, Microsoft azure account is created. Then an Azure web application and a SQL database is developed. Then the application is published to Microsoft Azure via Microsoft Visual Studio. Diagram 16 depicts the result when it is run on Azure, which implicates that the service is functional.



Moreover, MLSOA maturity model proposed by Dissanayake (2017) is displayed through the Machine Learning experiment shown in diagram 18. As it depicts, it imports data from the Azure database and carry out machine learning functionalities offered through Microsoft Azure Machine Learning Studio to identify the maturity level.



### Conclusions and Recommendations

This research paper proposes Service Oriented Architecture design to enhance business processes of British Gas. It proposes method for British Gas to stay competitive with current technology trends and enhance their service output to their customers. Initially, the research paper identifies functional and non-functional service requirements. Then functional descriptions were developed for the candidate service identification process. Moreover, following Thomas Erls' design principles, service task analysis and service classification were carried out, which allowed services to be separated into multiple categories. A use-case diagrams were developed to visualise sequence of actions provided through these services. A business process model was created for a selected service (create, update and delete customer). Moreover, MLSOA, which was proposed by Dissanayake (2017) was added into the overall business process to depict how British Gas can enhance their maturity level with SOA design principles. Furthermore, process simulations were carried out for modelled services and their efficiency were evaluated and analysed. Then SOA design for British Gas was developed and analysed through the design of UML component diagrams and SoAML diagrams designed following Thomas Erl's design principles. Finally, one functionality (create, update and delete customer) of the proposed SOA architecture was implemented as a WCF Service Application with .NET web application developed by using C# programming language. Then this service is moved onto Microsoft Azure cloud service to depict the efficient usage of acloud platform to provide service-oriented architecture to customers. The future work for this study involves the development of complete SOA application for all the requirements of British Gas. Moreover, the addition of MLSOA maturity model proposed by Dissanayake (2017) that was depicted in the BPMN model will allow the process improvement in the organisation. Thus, it should be fully developed with the aid of AZURE ML software. Overall, it concludes that this research paper proposes effective design principles and development methodology to implement a SOA for British Gas and enhance their business processes.

**REFERENCES**

- Ankolekar, A., Burstein, M., Hobbs, J.R., Lassila, O., McDermott, D., Martin, D., McIlraith, S.A., Narayanan, S., Paolucci, M., Payne, T., Sycara, K., (2002), DAML-S: Web Service Description for the Semantic Web, Proc. First Int'l Semantic Web Conf. (ISWC 02).
- Curbera, F., Duftler, M., Khalaf, R., Nagy, W., Mukhi, N., Weerawarana, S., (2002), Unraveling the Web Services Web: An Introduction to SOAP, WSDL, and UDDI. IEEE Distributed Systems Online 3(4): (2002)
- Delgado, A., García - Rodríguez de Guzmán, I., Ruiz, F., Piattini, M., (2010), BPMN business process models to SoaML service models: a transformation-driven approach. 2nd Int.Conf. on Software Tech. and Engineering (ICSTE 2010).
- Dijkman, R.M., Dumas, M., Ouyang, C., (2008), Semantics and Analysis of Business Process Models in BPMN. Information and Software Technology (IST)
- Dissanayake, S., (2017), The role of Machine Learning in Service-Oriented Architecture and Business Process Improvement. Research Paper, Leeds Beckett University, United Kingdom.
- Kobrym, C., (2000), Modeling Components and Frameworks with UML. Communications of ACM, 43(10); pp. 31-38.
- Donnelly, C. (2015), British Gas to power more of its business using public cloud from 2017 [Internet] Available from: < <http://www.computerweekly.com/news/4500248739/British-Gas-to-power-more-of-its-business-using-public-cloud-from-2017> > [Accessed on 01 April 2017]
- Elvesæter, B., Panfilenko, D., Jacobi, S., and Hahn, C. (2010). Aligning business and IT models in serviceoriented architectures using BPMN and SoaML. In Proceedings of the 1st International Workshop on Model-Driven Interoperability.
- Erl, T. (2005). Service-oriented architecture concepts, technology and design. Upper Saddle River, NJ: Pearson Education.
- Weinreich R., Sameting, J., (2001), Component models and component services: Concepts and principles, in: Component-Based Software Engineering. Putting the Pieces together, eds. G.T. Heineman and W.T. Councill, Addison-Wesley, New York, pp. 33-48.
- Malan, R., Bredemeyer, D., (2001), Functional Requirements and Use Cases. Research Paper, Bredyemer Consulting.
- Qian, L., Luo, Z., Du, Y., Guo, L., (2009) Cloud computing: an overview. In: CloudCom'09: proceedings of the 1st international conference on cloud computing. Springer, Berlin, pp 626-631.
- Ramachandran, M., (2016), Service Oriented Architecture (SOA) Introduction [PowerPoint Presentation], Leeds Beckett University. Available from: < <https://app.box.com/s/xzr4xgn1p71afp5icoc1> > [Accessed 03 April 2016].
- Sommerville, I., (2010), Software Engineering, 9th Edition, Addison Wesley.
- Tumay, K., (1996), Business Process Simulation. In J.M. Charnes, D.J. Morrice, D.T. Brunner, and J.J. Swain, editors, Proceedings of the 1996 Winter Simulation Conference, pages 93-98. ACM Press.
- Yigang, Z., (2012). Application of Zedgraph in Rapid Develop Graphic Chart, Research Paper, Beijing Institute of Aeronautical Materials, Beijing.
- Zhang, W., Cheng, G., (2009), A service-oriented distributed framework: wcf, in Web Information Systems and Mining, International Conference, pp. 302-305