TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

MUHAMMAD SOHAIL

CALCULATION OF ENERGY FOOTPRINT OF MANUFACTURING ASSETS

Master of Science Thesis

Examiner: Dr. Josè Martinez Lastra

Examiner and topic approved by the Faculty Council of Engineering Sciences meeting on December 4, 2013

PREFACE

The research work in this Master's thesis is carried out at *Factory Automation and System Technologies (FAST)* research group in the Department of Mechanical Engineering and Industrial Systems, Tampere University of Technology, Finland. The funding for research work came from EU project URB-Grade: *Decision Support Tool for Retrofitting a District, towards the District as a Service*.

First of all, I pay my deepest gratitude to Director FAST, Professor Dr. Josè Luis Martinez Lastra, for providing me with the opportunity to work in the diversified group of FAST and also for examining the work. Next, I would like to thank my mentor and supervisor, Anna Florea, from the depth of my heart for not only introducing me to this topic but also for her long term support, guidance and advice throughout the research work. It could not have been done without her support, supervision and patience with my work. I am also thankful to Dr. Corina Postelnicu and Dr. Andrei Lobov for their positive feedback and motivation they had provided me during my work. I am also thankful to all the personnel of the FAST group for providing me with such kind of conductive environment for the research work, especially my office colleagues and friends, Anton, Juha, Ahmed, Mehmud, Ville, Xu, Rajesh, Peyman, Luis, Sergii, for their kind support and helping hands.

I would also like to thank all my friends for their constant moral support and encouragement that I am able to perform this work. Special thanks to all my Pakistani friends in Tampere whose nice company, support and get-together never let me feel lonely and away from my home.

Finally, I would like to pay my deepest gratitude to my loving family especially my Parents whose kindness, care, encouragement and unlimited love helped me during my studies here and made me able to do this work.

I would like to dedicate my thesis to my niece Aizah Zareen and my nephews Abdullah Zareen and Abubakr Zareen.

Tampere, May 09, 2014

Muhammad Sohail

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master's Degree Programme in Machine Automation
SOHAIL, MUHAMMAD: Calculation of energy footprint of manufacturing assets.
Master of Science Thesis, 62 pages, 6 Appendix pages
May 2014
Major: Factory Automation
Examiner: Professor Dr. Josè Martinez Lastra
Keywords: Energy Management System, Energy Footprint, Key Performance Indicators Analysis, Data Analysis

Energy efficiency is one of the important topics in manufacturing sector, primarily due to increasing energy prices, ecological concerns and stringent regulations. Industries that have already spent many years in improving energy efficiency now find it difficult to propose and implement additional improvement measures. This thesis proposes a way to identify potential areas for improvements in energy efficiency by calculating energy footprint of the manufacturing assets. The information on energy footprint is provided using analytic tools. The main goal of the thesis is to demonstrate how energy footprint can help in making decisions regarding the manufacturing facility that may lead to improvement in the overall energy efficiency of the system.

For purpose of energy data analysis, a methodology is proposed in which the analytic tools are developed as web services. This methodology is suitable for implementation within a service-oriented manufacturing facility. The core analysis of data is carried out using MATLAB due to its powerful data analysis capabilities, high level programming, availability of relevant libraries, easiness in data handling and manipulation. MATLAB functionalities are deployed in Java and are published as web services. The classes in Java are mainly responsible for message parsing and sending/receiving data from MATLAB functions deployed in Java.

The work demonstrates the usefulness of analytic tools in analyzing energy footprint of manufacturing assets. The work also demonstrates applicability of service-oriented approach for data analysis and availability of analytic tools as web services in a service-oriented manufacturing system. The service-oriented approach enables extensibility of analytic web services and online availability.

This study results in development of analytic tools helpful in analysis of energy footprint of manufacturing assets within a service oriented manufacturing system. Energy footprint helps in identifying potential areas for improving energy efficiency.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

API	Application Programming Interface
ASF	Apache Software Foundation
EM	Energy Management
EMD	Environmental Management Dashboard
EMS	Energy Management System
ERP	Enterprise Resource Planning
EU	European Union
FASTory	Factory Automation and System Technologies Laboratory
FIL	Fujitsu Isotec Limited
GHG	Green House Gases
HTTP	Hyper-Text Transfer Protocol
ICT	Information and Communication Technologies
IDE	Integrated Development Environment
ISO	International Organization for Standards
JRE	Java Runtime Environment
JSP	Java Server Pages
JVM	Java Virtual Machine
KPI	Key Performance Indicator
MCR	MATLAB Compiler Runtime
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
UDDI	Universal Description, Discovery and Integration
UML	Unified Modeling Language
W3C	World Wide Web Consortium
WS	Web Service
WSDL	Web Services Description Language
WWW	World Wide Web
XML	eXtensible Markup Language

1. INTRODUCTION

Energy is the main driving force in nature. It exists in different forms, such as, heat, electricity, chemical, nuclear, etc. In the modern world, main consumers of energy are industrial sector, commercial sector, residential sector and transportation. Industrial sector consumes 51% of the total energy consumption in the world [1], which includes facilities and equipment used for agriculture, manufacturing, mining and construction. Within industrial sector, manufacturing activities account for around one third of world's total energy demand [2] and CO_2 emission [3] which makes it one of the biggest consumers of energy in the world. In Finland specifically, 46% of total energy is consumed by manufacturing, in 2012 [4].

Energy consumption in manufacturing is brought to the attention of the industrial companies due to number of factors, including increasing energy prices, ecological relevance and legislative pressure [5][6][7]. These factors have also challenged the manufacturing sector worldwide to improve is energy efficiency and reduce its CO_2 emissions by upgrading its technologies. For this purpose, European Union (EU) has set up the Strategic Energy Technology Plan (SET-Plan) [6] to bring about new energy strategies aiming to reduce EU global energy consumption and greenhouse gas emissions by 20%, by 2020.

Manufacturing companies have brought improvements in the energy efficiency of manufacturing processes in number of ways, for example, by improving setup time and cycle time of machines, process substitution, quality control systems [8] and by developing models of energy consumption of machines, measuring energy efficiency, and forecasting the energy consumption based on the developed models [9]. Intelligent manufacturing systems are also incorporated for improving energy efficiency by manufacturing companies and for reducing human errors and higher productivity.

Energy efficiency can be achieved by reducing consumption of energy while the production rates and comfort level of occupants in the facility are kept the same. So, industries are continuously looking for potential areas where improvement in energy efficiency is possible. Due to increasing range and variety of information available from the manufacturing system, it is quite challenging to get such information. So there is a need to identify and analyze the energy footprint which could help in identifying the potential areas where energy could be saved or, in other words, energy efficiency could be improved.

1.1 Background

Energy efficiency is one of the important topics in modern times and research is being done to address this issue by finding effective ways to continuously improve energy efficiency of manufacturing systems. The improvements are brought in energy efficiency ranging from devices at shop floor level to supply chain designing [10] at Enterprise Resource Planning (ERP) level. Attempts have been made to achieve energy efficiency by developing models of energy consumption of machines, measuring energy efficiency, and forecasting the energy consumption based on the developed models [9]. However, these approaches are system-centred so they only work for specific models based on specific static non-reconfigurable manufacturing systems, thus not solving the problem completely.

Industries that have already spent many years in improving energy efficiency now find it difficult to propose and implement additional improvement measures. One reason for this is energy management that needs to be refined. Energy management solutions have been developed by manufacturing industries which include processing and analysis of energy related data from the facility. Decisions are taken based on the results of data analysis. A cross-layer approach for implementing energy management in manufacturing is proposed [11] based on service-oriented architecture.

Data analysis tools are developed to analyze data and to extract useful information from the data. Help can be taken from these analytic tools to achieve the goal of improved energy efficiency. By utilizing the data analysis tools, it is possible to analyze and calculate energy footprint of assets within a manufacturing facility. Energy footprint identifies the energy utilization of assets and their intensity in manufacturing facility. Energy footprint shows how much energy is consumed by each asset or process. Impact of energy footprint on overall energy efficiency can be calculated using appropriate data analysis tools. Energy footprint helps user to better understand the distribution of energy use within the manufacturing facility and to compare the utilization and loss of energy. The energy mapping identifies the areas in which energy is utilized or lost. Element of an organization's activities, good and services that can affect the energy use is termed as '*Energy aspect*' [12]. Significance of an energy aspect can be identified by calculating its proportion in total energy use. The '*significant energy aspect*' has not only high proportion of total energy usage but it also has potential for more efficient energy use or offer high potential for achieving energy savings.

Significant energy aspects indicate the opportunities to improve efficiency by implementing best energy management practices, upgrading manufacturing assets or developing new technologies. Significant energy aspects can be identified based on energy footprint of manufacturing assets. Energy footprint provides a benchmark for evaluation of benefits of improving energy efficiency and for prioritizing opportunity analysis. The information obtained by calculating energy footprint helps taking important decisions

regarding the facility. These decisions lead to energy efficient systems, improved productivity and, as a whole, higher performance of the manufacturing facility.

1.2 Problem Definition

1.2.1 Justification of work

Increasing energy prices and stringent regulations have made energy efficiency an important issue in manufacturing systems. With expanding manufacturing facilities and process operations, the variety and range of data also increases which aid in obscuring the link between operations and energy consumption. With proper data analysis tools, the energy footprint of the manufacturing assets can be identified and analyzed which assist in taking necessary measures to improve energy efficiency and in defining and managing strategies to improve profitability.

Therefore, there is a need to identify and analyze the energy footprint of the manufacturing assets in a manufacturing facility using data analysis tools. These analytic tools help calculating energy footprint of the manufacturing assets which enable users to better understand the distribution of energy use within the manufacturing facility and to compare the utilization and loss of energy in the system. It also gives useful information indicating the potential areas where improvements could be brought, thus aiding in improving overall energy efficiency of the system.

1.2.2 Problem Statement

Problem addressed in this thesis is *"how to calculate energy footprint of manufacturing assets in a service-oriented manufacturing system"*.

1.3 Work Description

1.3.1 Objectives

Objective of this thesis is to identify and analyze energy footprint of manufacturing assets so that it can help in providing necessary information to identify energy aspects having high potential of improvements. The energy aspect that accounts for high proportion of total energy use and has potential for more efficient energy use is called the '*significant energy aspect*'. Improvements brought in the significant energy aspects will ultimately result in improving energy efficiency of the manufacturing system. Based on calculation of energy footprint of manufacturing assets, important decisions can be made, leading to improved overall energy efficiency of the manufacturing system.

1.3.2 Methodology

In order to achieve objectives of this thesis, following steps are taken:

1. Study of literature in domain of energy management and efficiency in discrete manufacturing systems. It includes study of established Energy Management Systems (EMS) and recognized standards.
2. Review of analytic tools and algorithms helpful in calculation of energy footprint of manufacturing assets. Review of available analytic tools and functions. It also includes review of analytic tools provided by various energy management solutions.
3. Review of possible approaches for data analysis in manufacturing systems. Study of information and communication systems in service oriented manufacturing system. Based on the study, proposing an approach for data analysis implementing Service-Oriented Architecture.
4. Study of data available about assets in manufacturing system. Relevant Key Performance Indicators (KPIs) related to the available data.
5. Selecting a set of data processing and analysis algorithms for development of analytic tools. Describing information needed and provided by each analytic tool.
6. Implementation of selected methodology based on service oriented approach. Development of analytic tools for calculation of energy footprint.

1.4 Thesis Outline

The rest of thesis is organized as follows: Chapter 2 introduces background of the thesis. It includes extensive literature review, theory of energy management system and service oriented architecture in context of discrete manufacturing systems. Chapter 3 explains the methodology of the thesis and also introduces the technologies used in implementation. Chapter 4 includes the Implementation of the atomic analytic web services along with the request and response objects. Chapter 5 documents results of the work followed by conclusions of the research work and recommendations for future research in chapter 6.

2. LITERATURE REVIEW

In context of discrete manufacturing systems, energy footprint provides necessary information to monitor and analyze the energy performance variables. These performance variables provide information about energy efficiency and related measures. For this purpose Energy Management Systems (EMS) are implemented in industries to monitor and analyze energy performance. Introduction to EMS and the established standards are provided in section 2.1 and in its subsections.

EMS uses data analysis tools for processing and manipulating data to present the data in informative way. The analytic tools are usually integrated within the application of EMS. For purpose of discoverability, accessibility and reusability of these analytic tools across the web for various applications, service-oriented approach can be implemented. Additional information on Service-Oriented Architecture (SOA) is provided in section 2.2 of this chapter. Benefits of implementing SOA are covered in subsection 2.2.4.

2.1 Energy Management System

Energy Management (EM) refers to set of measures designed and implemented with purpose of minimizing energy consumption while the comfort level of the occupants and production rates are maintained [13], whereas Energy Management System (EMS) is a computer-aided tool used for calculation, monitoring, and analysis of energy variables of a facility to provide basis for energy efficient decisions. EMS, when integrated in to organizational business structures, helps in saving energy, cost, and improves energy and business performance.

It is not enough to save energy only during the product utilization phase but the production phase as well. The term "*Energy engineering*" in manufacturing industries is an emerging topic due to economic and environmental reasons [15]. Strategies to reduce energy consumption in production include switching active assets into modes that require less energy input and other energy management practices.

EMS initiatives should be taken at component level in an industry. A systematic five steps approach is formulated to foster energy efficiency in manufacturing facility [16]. First to get knowledge of all the production process chain which includes information about production machines, material flow and production management. Next step is to perform detailed energy analysis of production equipment and also performing energy analysis of the building services. After these steps, analysis of load profile is performed to identify energy consuming components and specific characteristics of consumption; and the final step being the evaluation of production system. For improving energy effi-

ciency of factory equipment, visualization techniques for energy data are also proposed [17].

Energy management systems are implemented in many industries for purpose of monitoring and analysis of energy use. It helps the organization in planning and improving manufacturing processes. One good example is the EM solution provided by Honeywell [18], consisting of a customizable portfolio of new and existing hardware, software and services. It is designed to improve energy efficiency and reduce Greenhouse Gases (GHG) emissions. It includes a dashboard which gathers energy information of manufacturing assets and analyzes the data. It helps user to understand key energy indicators and their effect on overall energy consumption. In the same domain but a slightly different approach is followed by Fujitsu Isotec Limited (FIL). FIL has implemented Environmental Management Dashboard (EMD) [19] for accurate real-time visualization of energy consumption using Information and Communication Technologies (ICT). EMD divides the total energy consumption in to three broad categories: Energy used for facilities (fixed), energy consumed directly in manufacturing (consumption relative to production), and energy consumption varying with weather and seasonal changes (variable). This approach enables FIL to identify any abnormalities or impracticalities and also help identifying areas where energy is wasted. Awareness provided by such systems help organizations to take additional measures to save energy.

2.1.1 Standards

Planning to improve energy efficiency helps organizations to establish relevant objectives and identify and prioritize the areas or activities where there is potential to improve energy efficiency. This section briefly discusses about two EMS: ISO 50001 and EN 16001.

ISO 50001:2011

ISO 50001:2011 is standard created by International Organization for Standards (ISO) for specifying the requirements for establishing, implementing, maintaining and improving an EMS [20]. It was released by ISO in June 2011. It is an international energy management standard equivalent to European energy management standard I.S. EN 16001:2009 (details in next section) which was developed to help organizations to improve their energy efficiency in a logical, controlled and systematic way, to help saving energy and reducing costs [12].

This standard is based on management system model of continual improvement which forms the basis for other well-known standards ISO 9001 and ISO 14001. It provides a generic framework which is suitable for any organization, irrespective of the size, sector and geographical location.

ISO 50001:2009 framework helps the organization in many aspects, including:

- Developing energy efficiency policy.
- Setting targets and objectives to meet the energy efficiency policy.
- Using data to better understand the energy use and help make decisions accordingly.
- Measure the results and evaluating the policy.

ISO 50001:2011 like many other standards is based on Plan, Do, Check, Act (PDCA) cycle as shown in Figure 1.

Figure 1: The Plan, Do, Check , Act (PDCA) cycle[1]

.

Each element of PDCA cycle is briefly described below:

- ***Plan:*** Conducting energy review and establishing baseline, performance indicators, objectives and targets. Set action plans to achieve the targets and objectives.
- ***Do:*** Implementation of action plans set in planning phase.
- ***Check:*** Monitoring and measuring key indicators to determine improvements in the energy performance against the set targets.
- ***Act:*** Taking actions to continually improve energy performance.

The EMS model of ISO 50001:2011 is shown in Figure 2.

[1] Plan, Do, Check, Act Model [WWW] [Accessed 03.04.2014] Available on: http://en.wikipedia.org/wiki/File:PDCA_ZIRKEL_ENGL.png

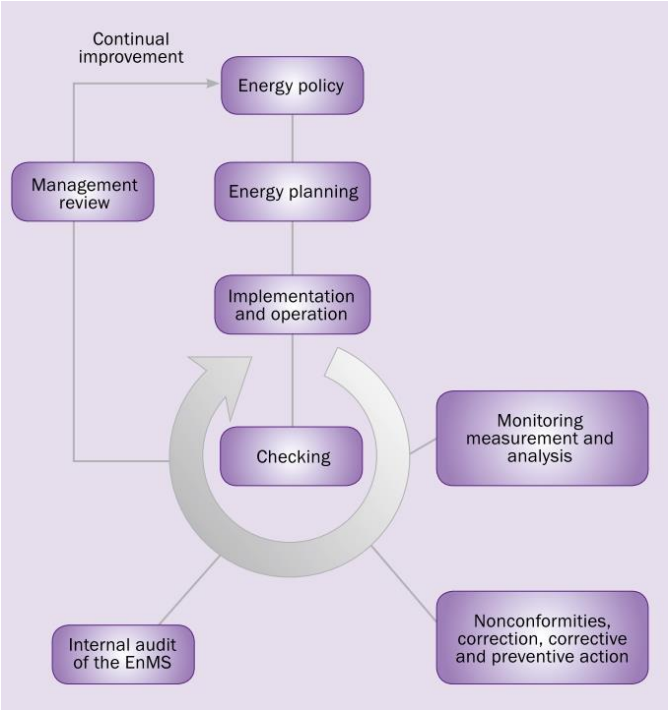

Figure 2: Energy Management Model of ISO 50001:2011[2]

EN 16001:2009

Like ISO 50001, EN 16001 is also a generic standard and can be applied to any organization irrespective of its size, sector or geographical location. EN 16001:2009 standard for EMS [12] documents detailed information about planning and implementing energy efficient practices in any organization. This standard broadly describes the EMS for planning energy efficiency in the following steps. First step includes identification and reviewing of energy aspects. It includes gathering energy consumption information (past and present) and presenting the patterns and trends possibly in the form of charts/graphs or tables/spreadsheets. The areas with significant energy aspects are identified based on available data using process maps, graphs, energy models, etc. Based on the available data energy consumption can be predicted for specific period of time. Figure 3 illustrates the review of energy aspects [21].

[2] Win the energy challenge with ISO 50001, [WWW] [Accessed on 03.04.2014] Available on: http://www.iso.org/iso/iso_50001_energy.pdf

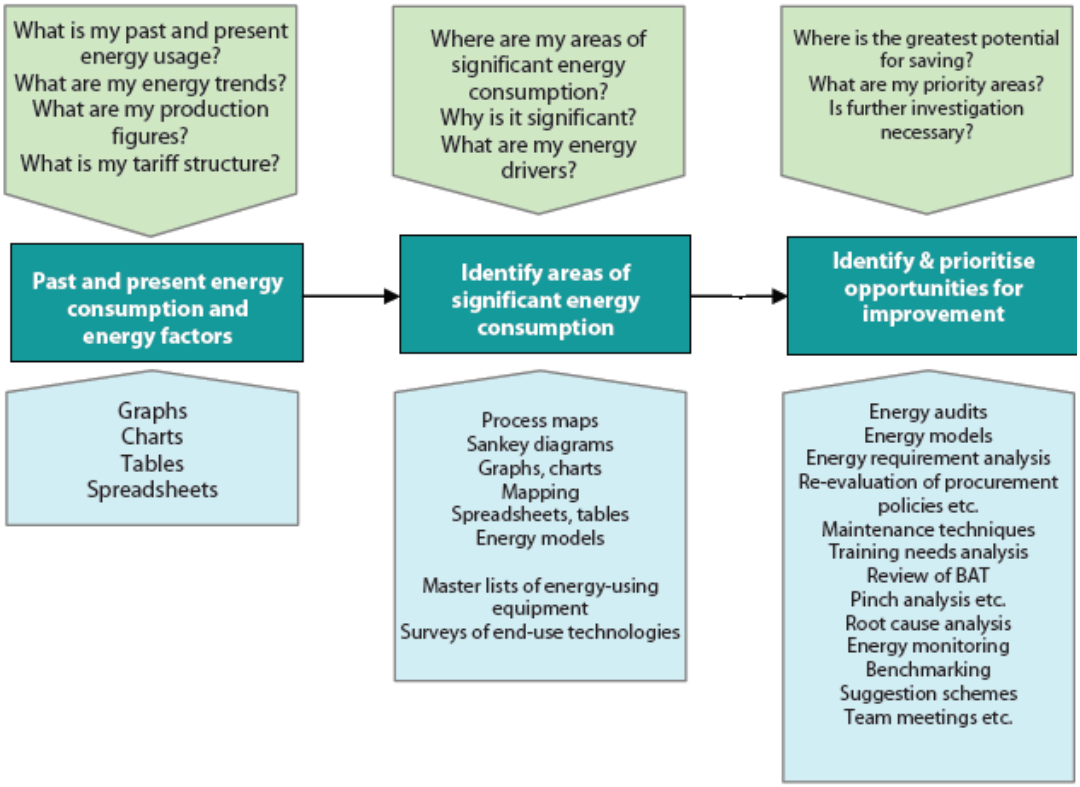

Figure 3: Review of Energy aspects [21]

Opportunities for improvement should be identified and prioritized continuously to bring continuous improvement in the organization. This could be accomplished using analysis tools for example: Pareto analysis, Root-cause analysis, System efficiency analysis, etc. Next step includes setting objectives and targets for energy consumption using techniques like regression analysis, CUSUM, Data mining, etc.

The implementation guide for this standard includes following steps:

1. ***Planning:*** It includes accessing current energy situation of the organization and setting goals and targets for energy efficiency.
2. ***Implementation and operation:*** It includes the measurements of resources, allocation of roles and responsibilities and authorities. It also covers various energy awareness techniques among the employees of the industries in form of training and competences.
3. ***Checking and corrective actions:*** It includes monitoring of energy variables and measurements related to energy usage and related factors. Internal audits for review of part or all of the organization is also included in these actions. Checking for non-conformity and planning corrective and preventive actions to avoid non-conformities.
4. ***Review, Managing and Improving EMS:*** This step helps in determining if the EMS implemented is suitable for the organization and to check the effectiveness of the overall system.

2.1.2 Energy Profiling and Analysis

Energy profiling or mapping is visualization of energy consumption and is used to disaggregate energy consumption into its constituent end uses. Information about energy consumption of each asset in manufacturing facility is required for energy profiling. ICT equipment (sensors network and energy meters etc.) provides to the system the energy related information about the devices and their states. For optimization of factory floor in terms of energy efficiency, emphasis is put on real-time monitoring of the energy related data from each machine [22]. Gathering the energy related data and analyzing for significance in specific context helps in making strategies for energy efficient system [22].

The value of continuous monitoring and analysis lies in benchmarking of energy consumption. Forecasting and analyzing helps in developing and implementing new policies, predictive maintenance and energy conservation measures. For purpose of data analysis, software tools can be used based on variety of programming languages. MATLAB is one such powerful tool for data analysis. It has the facility to deploy the functions or algorithms written in MATLAB in to other programming languages including Java and .Net. This attribute of MATLAB can be exploited by deploying data analysis algorithms and functions written in MATLAB in to Java and publish them as web services to be used in service-oriented manufacturing system.

In different domain, but mostly similar concept has been followed by Tomasz Markiewics [23] in remote pathology image analysis realized by MATLAB algorithms. The remote user can be connected with the system server via internet using a web browser. Tomasz Markiewics [23] utilized the image processing capabilities of MATLAB using Image Processing toolbox and compiled the MATLAB algorithms to executable jar files using the "MATLAB Builder JA" toolbox. Java Server Page (JSP) was used to make graphical user interface where user could provide the image to be analyzed along with the input parameters. The JSP page run using tomcat server. The client obtains the results as a graphical processed image. Additionally, using the MATLAB Database toolbox the processed images can be stored in the database.

Analysis of energy data can be done using analytic tools. Advantage can be taken from these analytic tools to analyze energy data and provide information in various aspects. Some of the tools that could be used in energy data analysis are Regression analysis, CUSUM analysis, Correlation analysis, and Sankey diagram [12]. With the help of these tools, potentially useful information can be derived which may help in identifying potential areas for improving energy efficiency.

2.2 Service Oriented Architecture (SOA)

Service-oriented architecture (SOA) is an evolution of distributed computing based on the request/reply design paradigm for synchronous and asynchronous applications. There are three main components involved in any SOA [24]: Service Consumer, Service Provider, and Service Broker.

- ***Service provider*** is an application providing the web service.
- ***Service consumer*** is an application using the web service.
- ***Service broker*** is an application used for service discovery.

Web services use Web Service Description Language (WSDL), Universal Description, Discovery and Integration (UDDI) and Service Object Access Protocol (SOAP) to communicate within a SOA [14]. A simplified diagram illustrating SOA is given in Figure 4.

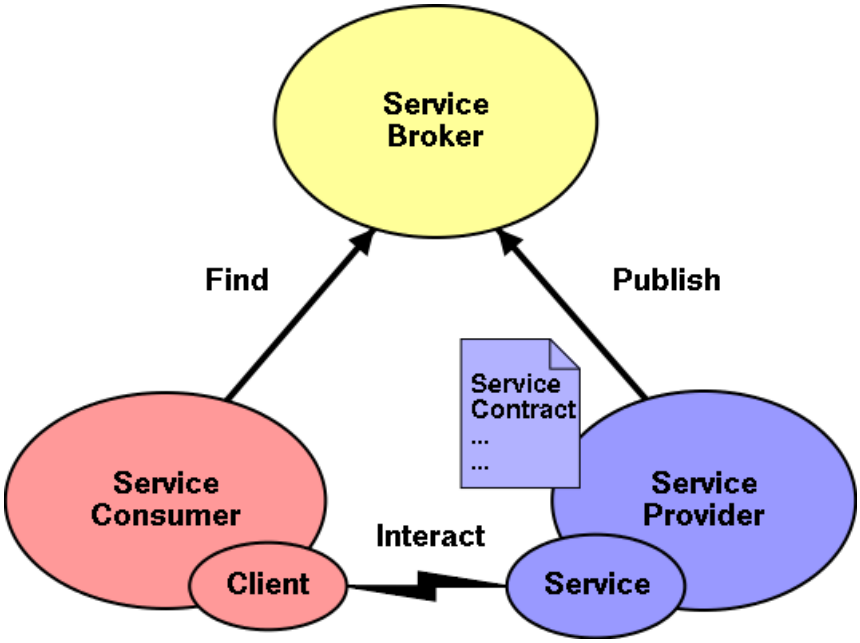

Figure 4: Main Elements in Service Oriented Architecture [3]

2.2.1 Web Services

Web services (WS) are self-contained, self-describing, modular applications that can be published, located and invoked across the web [25]. World Wide Web Consortium (W3C) defines WS as, "A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-process-able format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages,

[3] Service Oriented Architecture, [WWW] [Accessed 03.04.2014] Available on: http://www.w3.org/2003/Talks/0521-hh-wsa/slide5-0.html

typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards" [26]. WS are building blocks of SOA archetype. Web services are client and server applications that communicate over the World Wide Web's (WWW) Hyper Text Transfer Protocol (HTTP), as depicted in Figure 5.

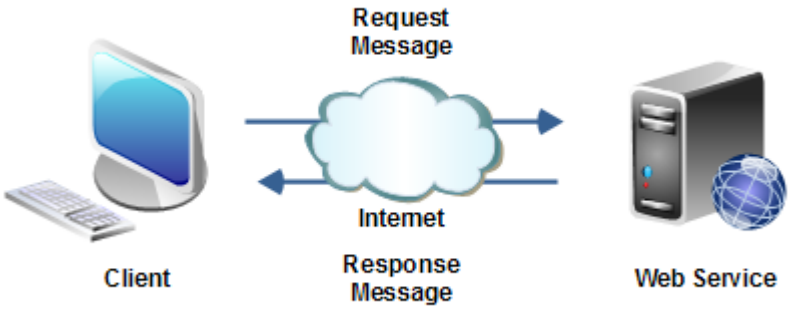

Figure 5: Client access web service through internet[4]

Web services provide a standard means of interoperating between software applications running on a variety of platforms and frameworks [27]. Web services can be combined in a loosely coupled way to achieve complex operations. Programs providing simple services can interact with each other to deliver sophisticated added-value services. WS can be reusable, discoverable, state-less and defined by contract.

Some of the properties of WS are given below:

- ***Self-Contained:*** WS are self-contained. This means that there is no additional software required on client side. On client side XML and HTTP client support, while on server side, a web server and a servlet is required. The implementation on client and server side can be completely in different platform or operating system.
- ***Self-Describing:*** For client to invoke any web service, it is enough to recognize only the format and contents of the request and response messages.
- ***Modularity:*** Web services can work independently in atomic form or they can also be aggregated to form more complex web services.
- ***Platform Independent:*** Since WS are based on concise set of open XML-based standards, they are platform independent. This promotes interoperability between clients and services across variety of platforms.

2.2.2 Fundamental principles of SOA

This section describes fundamental principles underlying SOA [28]. These principles form the basis of SOA and can be adopted by any organization regardless of type, geographic location or size of the organization.

[4] Web service message format, [WWW] [Accessed 07.04.2014] Available on: http://tutorials.jenkov.com/images/web-services/web-service-message-formats-1.png

Service abstraction

It is a design principle that allows developers of web services to control what part of underlying service logic are exposed to outside world. It enables to publish the services as black box by hiding the underlying details from the consumers. It is desirable because it supports loose coupling of the services (explained below).

Loose coupling

This design principle is incorporated in web services in order to ensure that the service contract is not tightly coupled to the service consumers, as well as to the underlying service logic and implementation. The connection between service consumers and service providers needs to be stable and well structured. So the loose coupling of services results in service contracts that could be freely evolved without affecting either the service consumers or the service implementation. A number of couplings can be established which includes, logic-to-contract, contract-to-logic, contract-to-implementation, consumer-to-contract coupling, etc.

Service Autonomy

Autonomy of services is essential to carry out their capabilities consistently and reliably. This demands that the underlying service logic should have significant degree of control over its environment and resources. Autonomy in the web services is of two types: design-time autonomy and run-time autonomy. Design-time autonomy is the independence with which the web services can be developed or evolved without affecting the service consumers. On the other hand, the run-time autonomy refers to the control limit a service has over its solution logic.

Service Reusability

Reusability of services is core of service-orientation. Reusability of services ensures the use of services by multiple applications across the enterprise. The design of services is made such that their solution logic is independent of any particular process or technology.

Service Statelessness

Scalable services are developed by separating the services from their state. It ensures minimization of resources consumption by avoiding the state data management. As a result, a service can handle more requests reliably.

Service Discoverability

Services can be discovered and interpreted using communicative informational metadata associated with them. Through Web Service Description Language (WSDL)

and service endpoint, user can access and interpret the information by the web service. Service registry with *Universal Description, Discovery and Integration* (UDDI) capabilities can improve service discoverability during runtime.

Service granularity

Granularity of a service defines the scope of the functionality in a service operation. A service can be expressed in relative terms of being either coarse-grained or fine grained service. Coarse-grained services have broader scope and functionalities but with more complexity, while the fine-grained services have narrower scope and relatively less complexity. The optimization of service scope is based on several parameters including performance, message size, transaction and business function, etc.

2.2.3 Web Service Development Approach

Approaches taken in development of web services have been categorized in to two [29]: Top-down (contract first) and Bottom-up (code first) approach. Description of each approach is given below.

Top-Down Approach (Contract first)

This approach involves creation of web services from a WSDL file (Details given in section 3.1.7) and supporting files. This is generally a recommended approach for making web services because it gives more hold or control over the WS. It reduces the interoperability issues that might come in using bottom-up approach. For purpose of this thesis, Top-Down approach is used in making the web services. A general diagram representing Top-Down approach for development of WS is given in Figure 6.

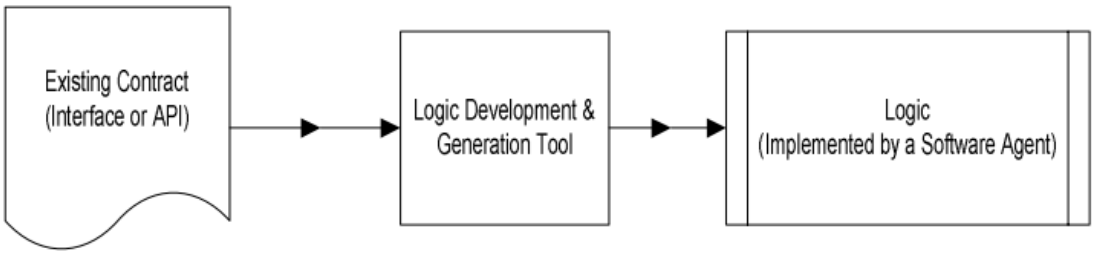

Figure 6: Top-Down approach for development of WS [29]

Bottom-Up Approach (Code first)

In this approach for developing WS, first Java bean is created (Interface class along with the implementation class), and then with the help of generating tool the WS is created. Although this method is faster and easier than the Top-down approach but it is not recommended because these may arise some interoperability issues. The general diagram representing Bottom-up approach for development of WS is given in Figure 7.

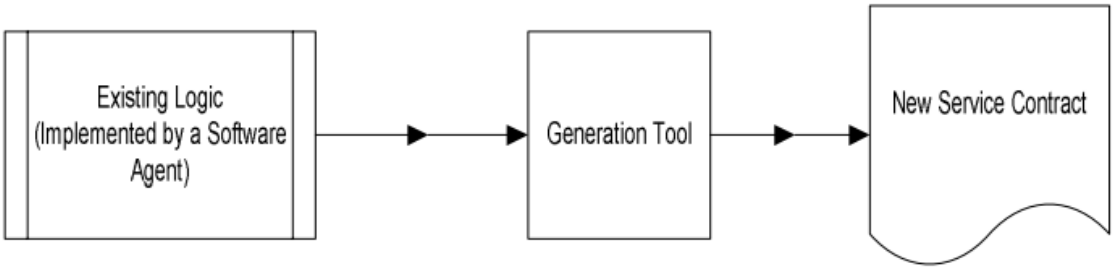

Figure 7: Bottom-Up approach for development of WS [29]

2.2.4 SOA in discrete manufacturing system

Within service-oriented manufacturing systems, every device interacts with outside world by providing and consuming information with other entities. Study [30] suggests that due to large number of devices interacting with each other, SOA seems a promising solution, in which each device offers its functionality as a service. So the future factories will be mainly based on SOA [30]. SOA paradigm is also needed for cross-layer real time information flow from shop floor to the Enterprise Resource Planning (ERP) system. One of the suggested possible approaches of integrating devices at shop floor within web services is by enabling the devices to run the web services natively so that each device offers its functionalities via a web service. It will make the integration of new devices in to the system painless and straightforward [30]. In order to achieve this, energy awareness should be introduced at the factory level using networked embedded technologies for automatic discovery of devices [31], for example, sensors/RFID readers, etc. In order to lead to energy efficient strategies, the data gathered at different layers (device level, location, process level to ERP level) should be correlated and commonly evaluated, including cross-enterprise issues [31].

SOA can be implemented for energy management in a manufacturing facility for monitoring and control of devices. By making each individual asset in the facility as a service using networked embedded devices and intelligent metering strategies (using wireless and indoor geo-location technologies), will help in monitoring of energy consumption resources regardless of the physical location of the service provider [31]. However, many other factors need to be considered before implementing SOA in any manufacturing system. An overview of factors effecting the implementation of SOA is given by [32] which indicate issues regarding to the organization, people, model and process, and maintenance. These factors should be considered in detail before implementing SOA in any system.

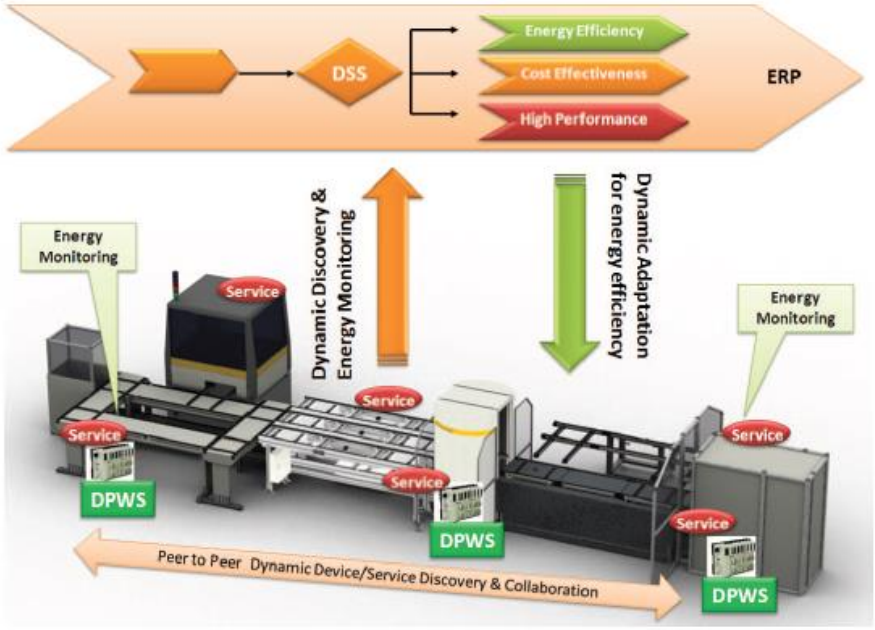

Figure 8: Service Oriented Production Line [31]

Figure 8 illustrates a service oriented production line in which each device (conveyor, controllers, robot cells, etc.) provides a service natively so that each device is continuously monitored of its energy consumption in real-time via a web service. It not only makes the integration of new devices in to the system straightforward but also the production line can be controlled and monitored of their functionalities (e.g. start, stop, stand-by etc.).

2.3 Technologies for data analysis

Many technologies are developed for data analysis based on popular programming languages, including C, C++, .Net, Java, C#, Visual Basic, etc. Each technology provides specific set of tools for data analysis and data mining with certain complexity and limitations. Some of the well-known technologies for implementation of data analysis are very briefly introduced below.

Weka

Weka[5] is open source software developed in java, for data analysis and mining [33]. It includes standard data mining and analysis tools for data pre-processing, classification, regression, clustering/association rules, visualization, feature selection etc. The advantages associated with Weka are: Free availability, portability due to completely based on Java programming, contain comprehensive collection of data mining and analysis tools, and easiness of use due to its user-friendly Graphical User Interface (GUI).

[5] Weka official webpage [Accessed 04.04.2014]: http://www.cs.waikato.ac.nz/ml/weka/

R

R[6] is an open source programming language and development environment dedicated for statistical computing and graphics. It contains data analysis tools, such as, time-series analysis, clustering, and classification. The capabilities of R can be extended by importing dedicated statistical analysis libraries to it.

MATLAB

MATLAB is a fourth generation programming language with extended capabilities of data analysis using statistical toolbox. It is high level programming language written in different environments including C, C++. More detail about MATLAB is given in section 3.1.1.

Microsoft Excel

Microsoft Excel[7] is an application developed by Microsoft Corporation, based on spreadsheet concept. It has the capabilities of data manipulation, statistical and graphical analysis supported by built-in functions. '*Visual basic for applications*' is a programming aspect for Microsoft Excel in which users can use variety of tools and numerical methods and return back the results in Microsoft Excel.

ROOT

ROOT[8] is a system or program containing set of object oriented frameworks, written in C++, for large scale data analysis. Apart from standard mathematical functions, these frameworks include functionalities, such as, regression analysis, matrix algebra, and multivariate data analysis tools.

SCaViS

SCaVis[9] is Java based interactive framework for data analysis and visualization. It has the capabilities of mathematical computation, data mining and data analysis on data. It contains 2D and 3D interactive data visualization feature. Analytic functions can be used using MATLAB or Octave syntax. Other functionalities include data clustering, neural networks, regression analysis, curve fitting, symbolic and numeric parametric computation.

---

[6] R [Accessed on 04.04.2014]: http://www.r-project.org/
[7] Microsoft Excel [Accessed on 04.04.2014]: http://office.microsoft.com/en-us/excel/
[8] ROOT [Accessed on 04.04.2014]: http://root.cern.ch/drupal/
[9] SCaViS [Accessed on 04.04.2014]: http://jwork.org/scavis/

ELKI

ELKI[10] is a Java based framework including data mining algorithms. ELKI is developed to allow and evaluate the combinations of arbitrary algorithms, data types, distance functions and indexes. It includes data analysis and mining algorithms, such as, data cluster analysis, anomaly detection, and spatial index structures. It also contains data visualization tools.

2.4 Key Performance Indicators

Key Performance Indicators (KPIs) are quantifiable performance measures which are used to evaluate performance or, in general, success of an activity or process. KPIs reflect the organizational goals, so the choice and selection of the KPIs are carefully made in order to achieve certain goals. KPIs can vary ranging from very simple to a very complex KPI. Compound KPIs are made from multiple atomic or base KPIs. KPIs are selected and prioritized according to the goals of the organization. As the goals of organization change, the selected KPIs and their priorities may also change along with it. In context of discrete manufacturing facility, KPIs are defined to evaluate the performance of manufacturing assets and processes. The KPIs may be related to energy, quality, economics, production processes, production states, management, and finance. Different KPIs are proposed for evaluation of energy performance [34]. Table 1 lists the energy related KPIs that are considered and analyzed in this thesis work.

Table 1: Energy related KPIs for discrete manufacturing system [35]

S.No	KPI Name	Description
1	Active Electric Power by Cells	Active power is true power consumed by an asset.
2	Root Mean Square Voltage	Equivalent DC voltage of an asset
3	Power Factor this month by Cells	Power factor is ratio of active power and apparent power expressed in percentage
4	Active electrical energy consumption by cell	Measure of active electricity consumption by each cell
5	Active Electric Energy consumption by Cell by process	Measure of active electricity consumption by each process within a cell.
6	Peak load	Maximum load of the manufacturing system
7	Process units by cell	Number of processes done per unit time by cell.

[10] ELKI [Accessed on 05.04.2014] : http://elki.dbs.ifi.lmu.de/

3. METHODOLOGY

This chapter gives overview of the tools and methods used in developing web services for data analysis. Figure 9 illustrates the tools and methods used in implementing the web services. For the purpose of data analysis, functions are developed in MATLAB. These functions are deployed in Java using MATLAB builder JA toolbox and MATLAB Compiler Runtime (MCR) is used to run the deployed functions. Overview of MATLAB software, builder JA toolbox and MCR is given in section 3.1.1 and in its subsections. Java platform is used for developing the web service implementation, interface, and the supporting classes and methods for parsing the messages. CXF technology is used in building the web services. Brief introduction to Java and CXF is given in section 3.1.2 and 3.1.3 respectively. Server is needed in publishing the web services to be available for clients to invoke and utilize. Tomcat is used as server, overview of which is given in section 3.1.4. The overview of supporting technologies in the web services, like eXtensible Markup Language (XML), Service Object Access Protocol (SOAP) and Web Services Description Language (WSDL) is given in the sections 3.1.5, 3.1.6 and 3.1.7 respectively. Figure 9 shows the implemented methodology (encircled in green).

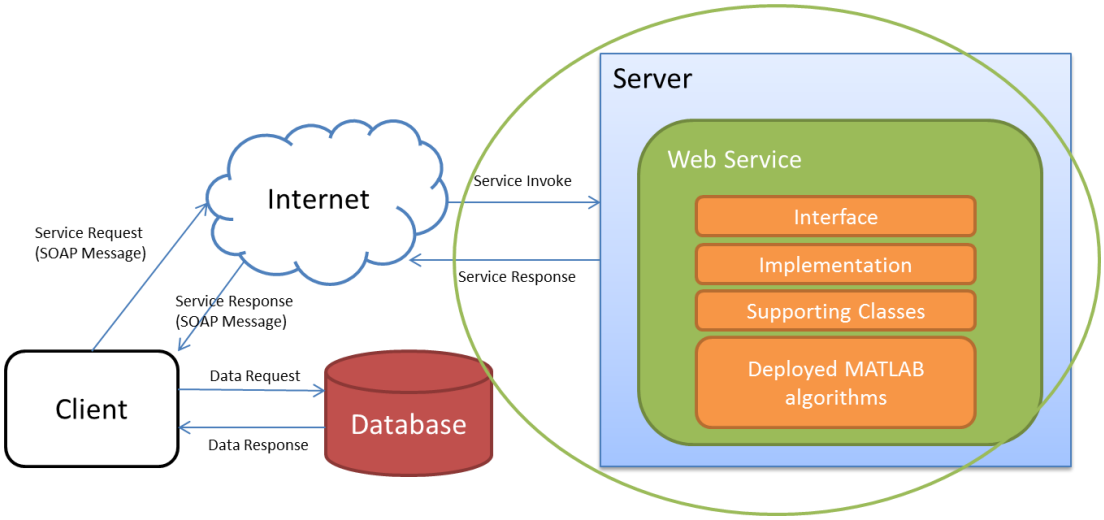

Figure 9: Overview of the implemented methodology

The server contains all atomic web services which can be invoked by client after providing the relevant data from the database. Implementation of this thesis work doesn't include making client and database, although each web service is tested by making individual client and providing data from manually created XML messages.

3.1 Tools and Methods

Table 2 summarizes the tools and technologies used in the implementation of this thesis work. Each technology is separately discussed in the following sections.

Table 2: Tools used in implementation

S.No.	Tool/Method	Version	Additional Info.
1	MATLAB	2013a	Used for data analysis
2	MATLAB Builder JA (Compiler)	4.18.1	Toolbox
3	MATLAB Compiler Runtime	8.1	
4	Java	SE 7	Eclipse Kepler IDE
5	Java Runtime Environment	7	
6	Apache CXF	2.7.10	Developing web services
7	Apache Tomcat	7.0	Server
8	XML	1.0	Message format
9	SOAP	1.1	Protocol
10	WSDL	1.1	Service Contract

3.1.1 MATLAB

MATLAB (Matrix Laboratory), a product of MathWorks[11], is a powerful numerical computing environment written in many languages, including C, C++, CIL (Common Immediate Language) , NVidia CUDA, Fortran and Java. It is fourth-generation programming language and is one of the highly developed software used in Engineering field for matrix manipulation, data or function plotting, making algorithms and making GUIs. It has the capability of interfacing with programs writing in languages including C, C++, Java, and Fortran. MATLAB also support for object oriented programming including classes, inheritance, packages, etc.

MATLAB offers a set of toolboxes which includes libraries and functions related to specific field of studies. It includes toolboxes from the family of Control System Design and Analysis, Signal Processing and Communication, Image Processing and Computer Vision, Computational Finance, Computational Biology, Code Generation and Verification, Parallel Computing, Database Connectivity and Reporting, etc.

For the purpose of making analytic functions, *MATLAB 2013a* is used in the implementation of this work along with all the necessary toolboxes.

MATLAB Builder JA Toolbox

Functions made in MATLAB can be deployed in Java using MATLAB Builder JA toolbox. This toolbox makes java classes from MATLAB programs which can be inte-

[11] MathWorks [Accessed on 14.3.2014]: http://www.mathworks.com/

grated into java programs. This toolbox encrypts the MATLAB programs and generates java wrappers around them which then behave like a common java class. This concept is illustrated in Figure 10. MCR is needed to run the compiled files (Details given in following section).

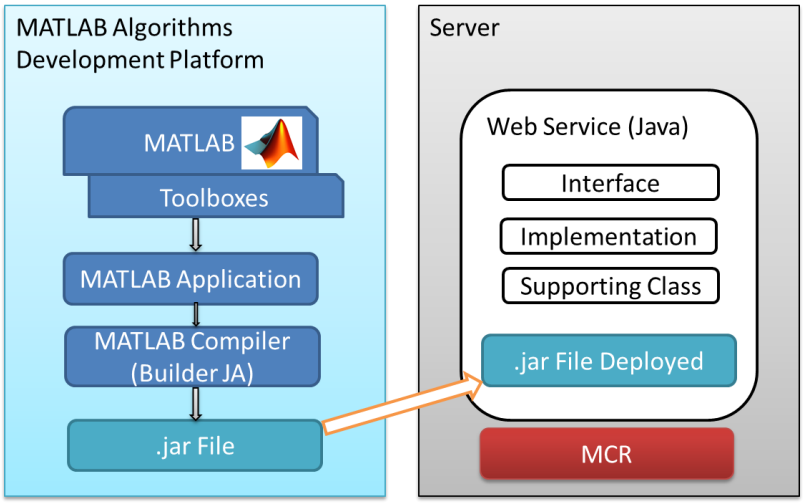

Figure 10: Wrapping MATLAB functions in java

These java programs can be deployed to any desktop application or web based server. The advantage is that the host computer only needs to have MATLAB Compiler Runtime (MCR) installed, even without having MATLAB installed in it.

MATLAB Compiler Runtime

MATLAB Compiler Runtime (MCR) is used to run programs which uses MATLAB compiled and deployed programs using MATLAB compiler. It is basically a standalone set of libraries that enables the execution of compiled MATLAB applications. The computers that do not have MATLAB installed in it can still execute the compiled programs. Figure 11 illustrates each of the builder products uses MATLAB compiler core code to create compiler components.

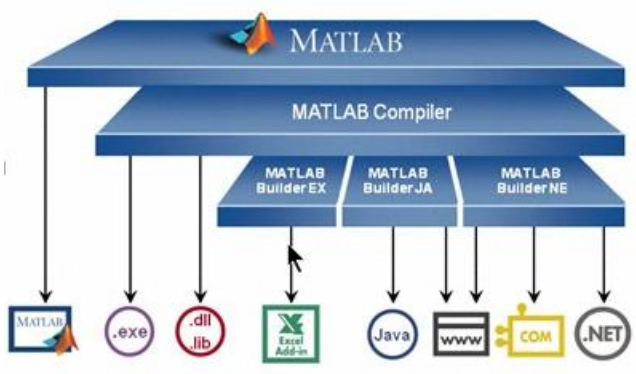

Figure 11: MATLAB application deployment products[12]

[12] MATLAB Compiler Runtime, [WWW] [Accessed 07.04.2014] Available on: http://matlab.fei.tuke.sk/applications/dotNetTech/moz.jpg

MATLAB 2013a used in the implementation of this work incorporates version *4.18.1* of *MATLAB Compiler* and it uses *MCR v8.1*.

3.1.2 JAVA

Java[13] is concurrent, class-based and object oriented programming language. It is platform independent language. Java programs are compiled to an intermediate representation called byte-code that can run on any Java Virtual Machine (JVM) regardless of computer architecture.

In order to write java programs, Eclipse[14] (Kepler Service Release 1) IDE is used in the implementation of this work. Java SE 7 (Update 51) and JRE7 are used throughout in the implementation process.

3.1.3 Apache CXF

Apache CXF[15] is one of the famous and widely used open source services frameworks. It helps in building and developing services using frontend programming APIs, like JAX-WS and JAX-RS. *Apache CXF 2.7.10* is used in the implementation process of this work. CXF provides various tools for generating code (wsdl2java, wsdl2js and java2js). It also supplies tools for generating WSDLs (for example: java2ws, xsd2wsdl and idl2wsdl). In this thesis work, wsdl2java functionality is used to generate code for web services by providing WSDL and request/response schema files. Figure 12 shows the Top-Down approach used for making web services using CXF 2.7.

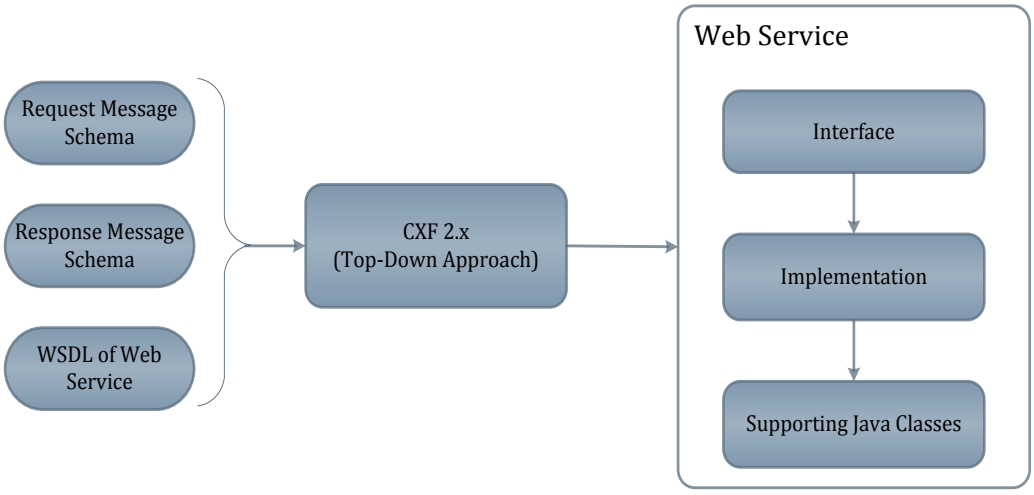

Figure 12: Top-Down approach for building Web Services using CXF 2.x

[13] Java [Accessed on 14.03.2014]: http://www.java.com/
[14] Eclipse IDE [Accessed on 14.03.2014]: https://www.eclipse.org/
[15] Apache CXF [Accessed on 14.03.2014]: http://cxf.apache.org/

3.1.4 Apache Tomcat

Apache Tomcat[16] or simply Tomcat, developed by Apache Software Foundation (ASF), is an open source web server and servlet container. Tomcat implements Java Server Pages (JSP) and Java Servlet specifications from Sun Microsystems. *Tomcat v7.0* is used as server in the implementation of this work. Figure 13 shows a java web service deployed on Tomcat server in this implementation.

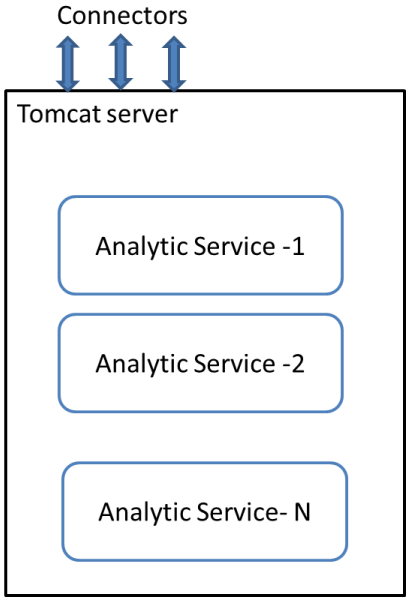

Figure 13: Java Web Services deployed in Tomcat server

3.1.5 Extensible Markup Language

eXtensible Markup Language (XML)[17] is a structured language design to structure, store, transport information. It is a markup language much like Hyper Text Transfer Protocol (HTTP). One of the advantages of XML is that it is independent of hardware, software, and application. XML is designed to be self-descriptive. It is W3C recommendation. General structure of XML message is given in Figure 14.

```
<?xml version="1.0" encoding="UTF-8"?>
<Element_Parent>
    <Element_Child1> ValueOfChild1 </Element_Child1>
    <Element_Child2 attribute2="AttributeValue">
        <Element_SubElement1> ValueOfsubElement1 </Element_SubElement1>
        <Element_SubElement2> ValueOfsubElement2 </Element_SubElement2>
    </Element_Child2>
</Element_Parent>
```

Figure 14: General structure of XML

"*XML 1.0*" version is used in making the request and response messages of the web services in the implementation of work.

[16] Tomcat [Accessed on 20.03.2014]: http://tomcat.apache.org/
[17] XML [Accessed on 20.03.2014]: http://www.w3.org/XML/

3.1.6 Simple Object Access Protocol

Simple Object Access Protocol (SOAP) is a standard format protocol specification for exchanging structured messages (communication) between applications via internet. SOAP is based on XML, so it is also platform independent, language independent, simple and extensible. SOAP is a W3C recommendation. Structure of SOAP message showing SOAP envelope, header and body is illustrated in Figure 15. *SOAP 1.1* is used in the implementation of this work for exchanges messages between the web services and the clients.

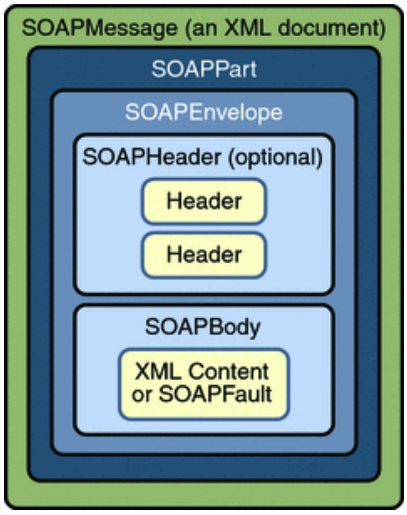

Figure 15: SOAP Structure[18]

3.1.7 Web Services Description Language

Web Services Description Language (WSDL) is XML based language used to describe and locate web services [36]. It consists of documentation, set of definitions to describe a web service in terms of location and interface (operations: Input and Outputs; and methods) of the web service. WSDL v2.0 is W3C standard. "*WSDL 1.1*" is used in the implementation of the web services for this work. A general structure of WSDL v1.1 is shown in Figure 16.

[18] SOAP [Accessed 20.04.2014]:http://docs.oracle.com/javaee/5/tutorial/doc/figures/saaj-noAttach.gif

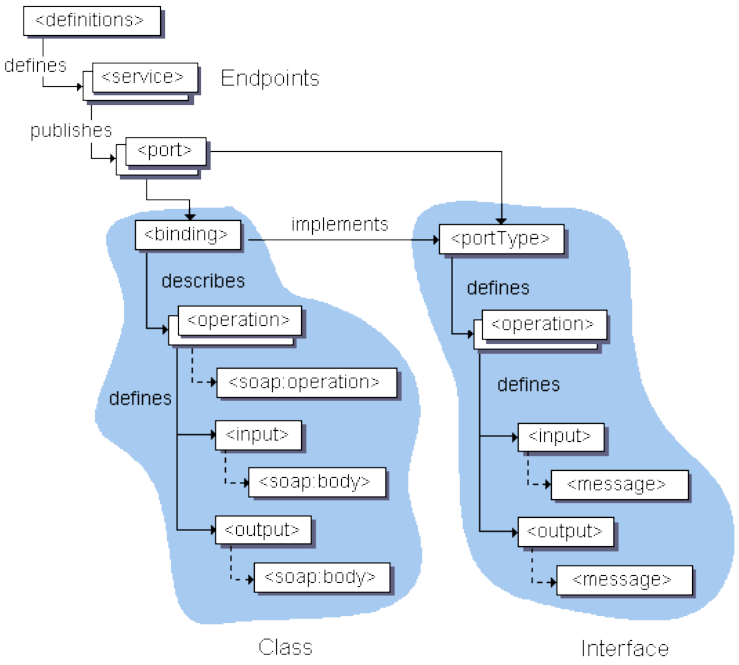

Figure 16: General Structure of WSDL v1.1 Document [19]

As shown in the figure above, WSDL Document has a root element ***"definition".*** The corresponding elements inside the root elements are described below:

- ***Types:*** It is a container for data type definitions used by messages.
- ***Message:*** It is an abstract, typed definition of the data communicated in a call.
- ***Port type:*** It defines collection of operations. The name of the operation is the name of the method being called.
- ***Operation:*** It defines the combination of input (message sent to the server), output (message which is sent to the client) and fault (error value when some problem occurs in processing of message) messages.
- ***Binding:*** It describes the protocols and data format specification by one or more endpoints.
- ***Port:*** A single endpoint specifying address for a binding.
- ***Service:*** Defines the collection of related endpoints.

The WSDL document which describes a web service acts as a contract between server and client. Following this contract, client and server can exchange information and data regardless of underlying platform or operating system on which they are operating.

[19] WSDL, [Accessed 03.04.2014] Available on: http://download.oracle.com/otn_hosted_doc/jdeveloper/1012/web_services/images/wsdl.gif

3.2 Deploying MATLAB functions in Java Applications

MATLAB Builder JA product is used to create a project, which enables the MATLAB functions or scripts to be used in Java applications. This product supports data conversion between Java types and MATLAB types.

Components created by the MATLAB builder JA are standalone Java packages (.Jar files). Each package can contain multiple Java classes encapsulating the MATLAB functions serving as the methods that can be called from Java. Following steps are necessary to take in order to deploy and access MATLAB functions from Java.

3.2.1 Writing MATLAB functions

MATLAB functions can be written by creating new function within MATLAB editor. The function name written in MATLAB serves as method name within the Java. The inputs and outputs for the function are defined explicitly along with their data types because data needs to be sent to and received from these functions.

3.2.2 Compiling MATLAB project

When the function is written and ready to be compiled, the MATLAB Compiler is invoked. The project name needs to be defined at this place. For compiling the project for Java, the 'Java Package' option needs to be selected as shown in Figure 17. MATLAB also offers compiling in other languages including C++ and .Net.

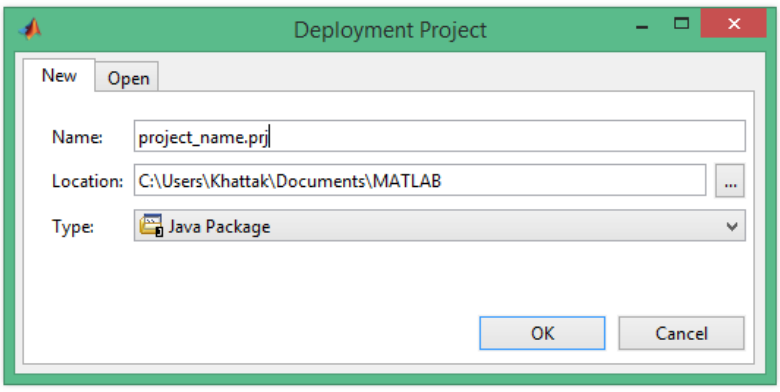

Figure 17: Assigning project name to the deployment component

After making the package, names of the classes need to be defined. The class name serves as a Java class within Java program. The functions are added to the corresponding class which serves as methods of the corresponding Java class. In the Project setting it is also possible to explicitly choose the toolboxes which are used in the functions to be compiled, as shown in Figure 18.

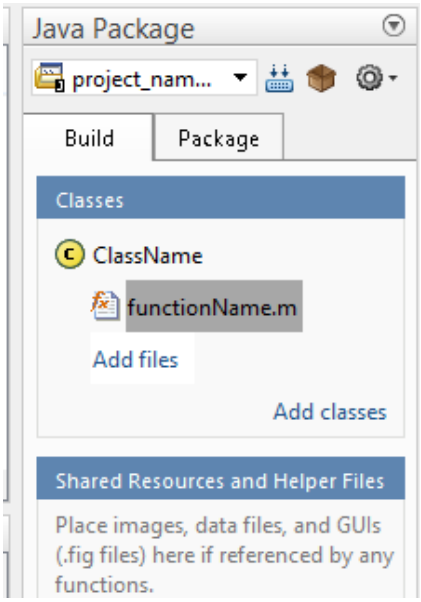

Figure 18: Defining Classes and Methods with corresponding toolboxes

Additional supporting files can also be added at this stage. When all the functions are added to the respective classes, the project is built. It results in creation of the project folder containing the following:

- *Distrib* folder: containing distributable jar file of the compiled project, along with the supporting documentation and *Read Me* file.
- *Src* folder: containing source files in addition to the files included in the *distrib* folder.

Compiler creates several overloaded methods when it processes the MATLAB code. The overloaded methods implement the MATLAB functions and each of these methods corresponds to the generic MATLAB function with a specific number of input arguments. Also, the compiler creates another method that defines the return values of the MATLAB function as an input argument.

3.2.3 Accessing MATLAB functions in Java

The compiled Java package (.Jar file) can be added in to the build path of the java program which needs to utilize the Java package. It is also necessary to add the java builder package (.Jar file) to the path, as the Java program needs use the *MWArray* classes.

When the Java package is imported to a specific class, all the classes and associated methods can then be utilized similar to any other Java class. However, there is one exception from the conventional method calling procedure. The first argument in the input arguments of a method must always be an integer, representing the number of expected outputs. For example: A function made in MATLAB takes three input arguments V, W and X, and return two output argument Y and Z. While calling this function, four (not three) arguments need to be provided within the Java program. First argument is the Integer representing the number of expected outputs, which in this case is two (Y and

Z), while second, third and fourth arguments will be the first (V), second (W) and third (X) input arguments respectively. It is important to convert the Java data types to MATLAB data types when sending input arguments to MATLAB function. The details of inter-conversion rules for data types are covered in section 3.3.2.

3.3 MATLAB Builder JA API

MATLAB builder JA toolbox provides an Application Programming Interface (API) to enable Java applications to exchange data with MATLAB methods. This API is implemented as *com.mathworks.toolbox.javabuilder.MWArray* package. Details of *MWArray* and associated array types are covered in the following sub-section.

3.3.1 MATLAB Array types

The root of the data conversion class hierarchy is the *MWArray* abstract class. Following are the subclasses of the MWArray class representing major MATLAB types:

- *MWNumericArray*: Base class for all numeric MATLAB array types.
- *MWLogicalArray*: Manages a native MATLAB logical array.
- *MWCharArray*: manages a native MATLAB char array.
- *MWCellArray*: Manages a native MATLAB cell array.
- *MWStructArray*: Manages a native MATLAB structure array.
- *MWFunctionHandle*: Represents MATLAB function handle.
- *MWJavaObjectRef*: Used to create MATLAB array that reference a Java Object.

Each of these classes provides constructors and methods associated with each class. These methods can be used to access the underlying array properties as well as the data. These classes, or in general the MWArray class, provide the following:

- Constructors for initializing the MATLAB arrays
- Finalizers for disposing the MATLAB arrays.
- Getter/Setters to read/write the array data.
- Methods to identify properties of the array.
- Comparison and conversion methods.

3.3.2 Inter-conversion between Java and MATLAB data types

Data types handled by MATLAB are different than those of Java. Therefore, it is necessary to convert Java data types to MATLAB data types when calling MATLAB deployed functions. Similarly, it is also necessary to convert the MATLAB data types to Java data types when results are returned by MATLAB deployed functions. The conversion rules apply to scalars, vectors, matrices and multidimensional arrays. The conversion rules from Java to MATLAB for some of the data types are given in Table 3.

Table 3: Java to MATLAB Conversion rules

Java Type	MATLAB Type
double	double
float	single
byte	int8
int	int32
short	int16
long	int64
char	char
Boolean	logical
java.lang.Double	double
java.lang.Float	single
java.lang.Integer	int32
java.lang.Byte	int8
java.lang.String	char

The Table 4 lists the data conversion rules for converting MATLAB data types to Java types. It is to be noted that Java has no unsigned types to represent the uint8, uint16, uint32, and uint64 types used in MATLAB. Similarly, the cells and structures in MATLAB are returned as Object in Java, and there are no Primitive Java data types for both of them.

Table 4: MATLAB to Java Conversion rules

MATLAB Type	Java Type (Primitive)	Java Type (Object)
Cell	Not applicable	Object
Structure	Not applicable	Object
Char	char	java.lang.Character
Double	double	java.lang.Double
Single	float	java.lang.Float
int8	byte	java.lang.Byte
int16	short	java.lang.short
int32	int	java.lang.Integer
int64	long	java.lang.Long
uint16	short	java.lang.short
uint32	int	java.lang.Integer
uint64	long	java.lang.Long
Logical	boolean	java.lang.Boolean

4. IMPLEMENTATION

This chapter gives details of the implementation of each analytic web service created in this thesis work. Details about web service implementation are covered in section 4.1. Section 4.2 briefly explains how the MATLAB deployed functions are accessed from Java application and how the images are returned back to Java. Structure of the web projects is covered in section 4.3, followed by details about functionality and application of each analytic web service in section 4.4. In the last section (section 4.5) the web service objects representing the request and response messages are discussed.

4.1 Web service Implementation Overview

Web services are published using Tomcat server. Each web service can be invoked by client using SOAP containing XML message implementing XSD given in Appendix A. The XML message is parsed by Java class and sent to the deployed MATLAB function for analysis. MCR is initialized when a web service is invoked, or alternatively MCR can also be initialized when the Tomcat server is started. The deployed MATLAB functions returns back the result to the java class which copies the results to the XML response message according to the XSD given in Appendix B. The response XML message is sent back to the client using SOAP. This implementation is illustrated in Figure 19.

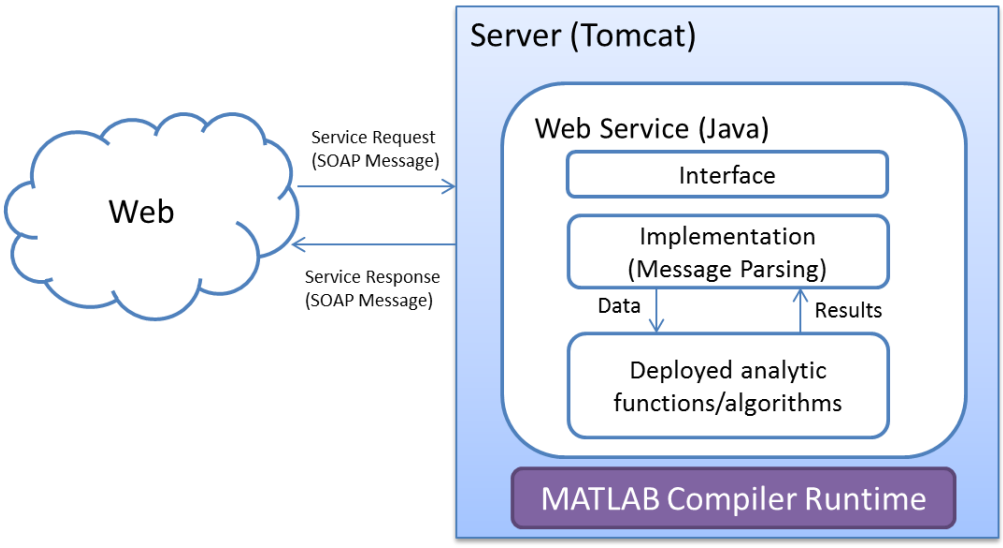

Figure 19: Implementation of Web Services

Sequence diagram of invoking the web service by client and getting back the results is shown in Figure 20. The client first invokes the web service by providing relevant data inside the SOAP message. The Java implementation class parses the message and assigns the data values to local variables. The variables are then passed to Service Class in which all the constructors are defined. The Service Class send convert the variables from Java type to MATLAB supported types using *MWArrayFunctions* Class. *MWArrayFunctions* contains methods to convert many java data types in to MATLAB data types and vice-versa. After converting the data to MATLAB supported MWArrays, the deployed function of MATLAB is called. Instantiating MATLAB deployed function also initializes MCR. The compiled jar file manipulates the data and gives back the results to the Service Class. The Service class again calls *MWArrayFunctions* Class for converting back the data to java supported data types. After converting the data, the Service Class returns back all the result values to service implementation class. The service implementation copies all the results in to return XML message and the service sends back the response message as SOAP to the client.

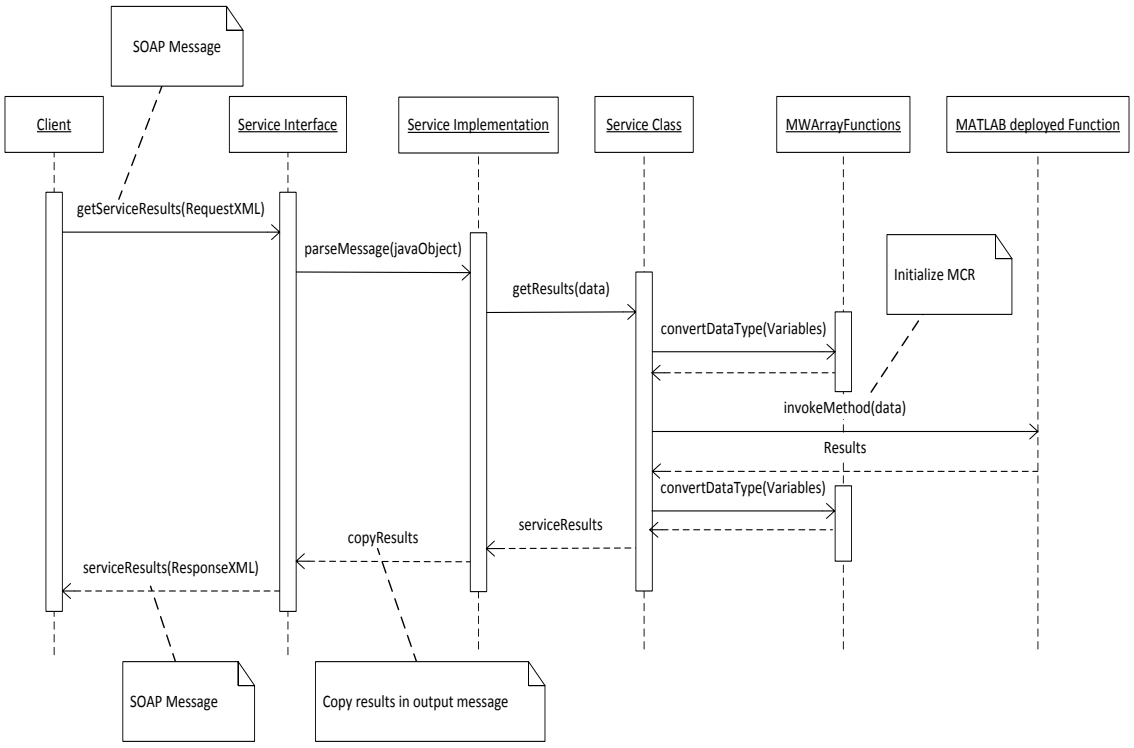

Figure 20: Sequence diagram of web service utilization

4.2 Calling MATLAB deployed functions

Each web service has a separate class which includes constructors and methods related the web service and calls MATLAB functions deployed in Java with the necessary data. The snippet below shows the correlation web service class calling the MATLAB deployed function.

```
//Importing java builder APIs
import com.mathworks.toolbox.javabuilder.MWArray;
import com.mathworks.toolbox.javabuilder.MWClassID;
import com.mathworks.toolbox.javabuilder.MWNumericArray;
//Importing MATLAB deployed project
import analysis_correlation.*;


public class DataCorrelation {
	private Double[][] inputValues = null;

	public DataCorrelation(Double[][] inValues){
		this.inputValues = inValues;
	}
	public Double[][] correlateData(){

	Object[] result = null;
	CorrelateData newData = null;
	Double[][] resultData = null;
	//Converting Java data type to MWumeric Array
	MWNumericArray inputData = new MWNumericArray(inputValues, MWClas-
sID.DOUBLE);

	try{
	//Instantiating the MATLAB deployed Class
	newData = new CorrelateData();
	//Calling method of MATLAB deployed Class and storing the result
	result = newData.correlate(1, inputData);
	//Instantiating MWArrayFunction Class to covert data type
	MWArrayFunction newFunction = new MWArrayFunction();
	//Converting the result to Double[][]
	 resultData = newFunction.NumericArrayObjectTo2DByteArray(result[0]);

	  }
	  catch(Exception e){
		e.printStackTrace();
	  }
	  finally{
		//Disposing the MATLAB class instant and result object array
		newData.dispose();
		MWArray.disposeArray(result);
	  }
	  return resultData;
	}
```

It can be seen from the above snippet of code that the Java class converts the local variables to *MWArray* types before passing them to the MATLAB functions. In the above figure, *inputValues*, which is a two-dimensional Double array of Java, is first converted to *MWNumericArray* data type by explicitly specifying the Class ID as Double. It can also be seen that response from the MATLAB function is stored in *result* variable which is an object array. The Object array, *result*, is then converted to two-dimensional byte array of java by instantiating and calling method within MWArray-Functions class explained in previous section.

It is also essential to dispose the arrays to free the memory and ensuring fast operation of the function. In the above code snippet, the variable '*result'* which is a byte array has been disposed before returning the result. The instant of MATLAB class '*CorrelateData'* has also been disposed before returning the result.

MWArrayFunctions Class

In implementation, the conversion of data types from/to Java to/from MATLAB is required extensively. Therefore, a separate Java class "*MWArrayFunctions*" has been created which includes all the methods required to convert Java types into MW Arrays and vice-versa, as shown in code below. This helps in quick conversion of data types without repeating the same code in various classes of analytic web projects.

```
import com.mathworks.toolbox.javabuilder.MWNumericArray;
//Convert List<Float> to MWNumericArray
public MWNumericArray ListFloatToNumericArray(List<Float> inList){
	Float[] tempArray = new Float[inList.size()];
	tempArray = inList.toArray(tempArray);
	MWNumericArray outArray = new MWNumericArray(tempArray);
	return outArray;
}

//Convert Object of MWArray (Numeric) to List<Float>
public List<Float> NumericArrayObjectToListFloat(Object inObject){
	MWArray tempArray = (MWNumericArray) inObject;
	List<Float> outList = new ArrayList<Float>();
		for (int index =1; index <= tempArray.numberOfElements(); in-
dex++){
		outList.add(Float.parseFloat(tempArray.get(index).toString()));
	}
	return outList;
}

//Convert Object of MWArray(Numeric) to Byte Array.
public byte[] NumericArrayObjectToByteArray(Object inObject){
	MWNumericArray tempArray = (MWNumericArray) inObject;
	byte[] outArray = tempArray.getByteData();
	return outArray;
}
```

It can be seen from the above snippet that Java-Builder library is needed to convert to/from MWArray data types.

Returning Image data

In the developed analytic web services, some of the services return image as the output. These images are returned from MATLAB to java in form of byte arrays. It is accomplished by using *figToImStream* function of MATLAB. This function basically steams out the snapshot of the MATLAB figure as byte array encoded in specified format by creating signed byte array. The image formats currently available through this function are: PNG, JPG, BMP, and GIF. The output type can either be int8 or unit8. Int8 is used primarily for java.

The byte array is encoded in to string before it is sent back as service response. The encoding of byte array in to string is carried out using Base64 encoding. The string is copied in the output message and is decoded on client side using Base64 decoder.

4.3 Structure of web projects

There is up to certain extent similarity in structure of every analytic web service made in this thesis work. Generally, every web service consists of following packages:

1. Request Object Package
2. Response Object Package
3. Web Service Package, containing
    a. Web service Interface
    b. Web service Implementation
    c. Service Class
    d. MWArrayFunction Class
4. Library folder, containing
    a. Java-Builder library
    b. Compiled MATLAB project

The above mentioned packages are illustrated in the Unified Modelling Language (UML) package diagram shown in figure below.

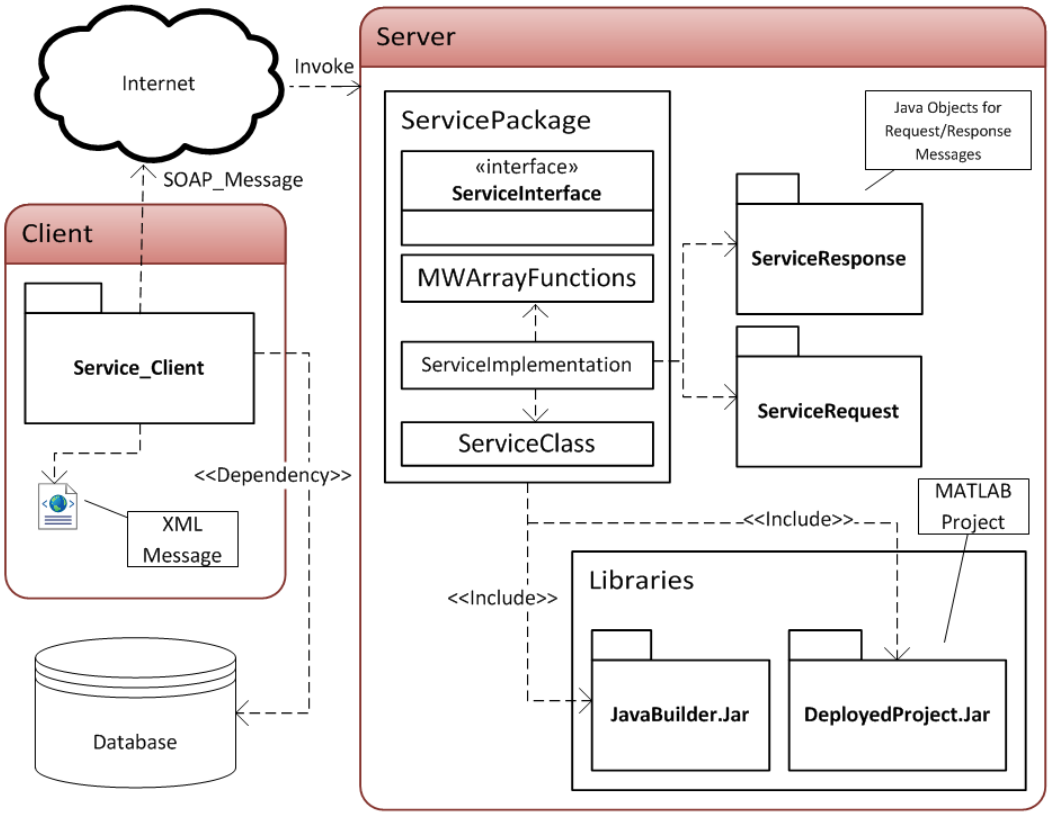

Figure 21: UML Package diagram of the implementation

As the names suggest, the Request and Response object packages contain the Java objects fully describing and representing the request and response messages of the web services. Each element, metadata, and associated data values are represented by a Java class for both request and response in the respective packages. These two packages are

same for every web service because all the web services are designed according to the same request and response schema.

The web service package contains the service interface and implementation of the respective web service. It defines the methods, input and outputs of the web service. The implementation class is used mainly to parse the incoming message and send the parsed data to the service class which contains the constructors and methods for calling the MATLAB functions. This package also contains the *MWArrayFunction* class whose function is to convert the data types between Java and MATLAB and vice-versa. The results returned by the service class are sent back to the implementation where the results are copied in the output message and returned to the client.

Each web service also consists of a set of libraries which essentially contain *Java-Builder* library and the compiled MATLAB project. The Java-Builder library contains the classes for all *MWArrays* needed in the web service. The compiled MATLAB project library contains the classes and methods that were written and compiled in MATLAB using MATLAB Compiler.

The MATLAB functions deployed in Java using *BuilderJA* toolbox are placed in *lib* folder of WEB-INF. It is essential to put them in this folder and not on local computer folder because the web service after publishing has to access these packages. An essential library of *Java-Builder* also needs to be placed in *lib* folder because it is used by the classes inside the web project.

4.4 Web Services

This section contains the description, functionality and application of each atomic analytic web service implemented for helping in analyzing energy footprint of manufacturing assets.

4.4.1 Data Consistency

Any analytic web service requires data from the database to be processed and analyzed. Any missing values in the database may result in giving falsified/inaccurate results. For example: The database might have some missing values of KPIs corresponding to some timestamp, or even missing the whole row of information along with the timestamp. It is therefore very important that the missing information or data must be identified and evaluated first in order to make sure the data is complete, and has no missing values, before sending it to any of the analytic service.

Functionality

For purpose of data consistency, a MATLAB function is made. The built-in functions for interpolation and extrapolation in MATLAB are used for this purpose. The list of interpolation methods available for estimation of missing data is follow:

- Method 0: nearest neighbor interpolation

- Method 1: Linear Interpolation and Extrapolation (Linear)
- Method 2: piecewise cubic spline interpolation and extrapolation (Spline)
- Method 2 is the Default method if not provided by the user.
- Method 3: shape-preserving piecewise cubic interpolation and extrapolation
- Method 4: Cubic Interpolation and Extrapolation
- Method 5: Polynomial Interpolation

Raw data coming from different sensors is stored in the database. The timestamps for these data are usually stored in the database in resolution of milliseconds (as floating points). There is a possibility that these stored timestamps do not have exact intervals. For example, if the frequency of stored data from a sensor X is 1 data per minute, then the interval between any two time stamps should be exactly (=1x60x1000) 60000 milliseconds. But in reality there might be variation of, say, 100 or 500 milliseconds. Such variations in the timestamps must be taken care of; otherwise the function would assume that there is a missing value and will insert an interpolated data in that place, which will result in exceeding amount of irrelevant data points. For this purpose, a variable is introduced in this function which accounts for percent interval variations. The default value of 5% is used, if user doesn't change it. It means that by default, 5% of variation in the intervals between the time stamps is tolerated. In the above example, 60000 milliseconds of timestamp interval would have tolerance range of +/- 3000 milliseconds (or +/- 3 seconds).

One other possibility is that the data has no missing information. In that case, the original data is sent back in the response object. It is best practice to invoke this web service of filling gaps before applying any other analytic tool on the data.

Application

This web service can be used in following applications:

- Checking for missing values in the data to be analyzed.
- Checking for null values
- Interpolating missing values by different available methods
- Returning back completed data.

4.4.2 Data Normalization

Normalization, in general, means adjusting the data values that are measured on different scales in to a notionally common scale. Normalization is performed on data for many reasons depending on the application. Most commonly, it is used for eliminating or minimizing dependencies, or applications where data measured in different units are to be plotting or analyzed together. In the later application, normalization of data involves scaling of data between ranges of values. One example of analysis where user can use normalized values is making Radar Chart (or Spider Web Chart; details given in

section 4.4.3) where different KPIs are plotted on a same polar coordinates chart. The normalized values enable the comparison of those KPIs although the units of each KPI differ from the rest.

Functionality

For normalizing data, a function in MATLAB is made, which is also integrated in to java, and published as a web service. This web service takes the data to be normalized, along with the range between which the data is to be normalized. If the range is not provided by the user, then the default range of {1-10} is used. The normalized values are returned in the returned object of the web service. It is essential that user provides at least three data points for normalization, otherwise, there is no logic of normalizing data having less than 3 data points. Formula for an array of values X between minimum value *a*, and maximum value *b* is given below.

$$X' = a + \frac{(X - Xmin).(b - a)}{Xmax - Xmin}$$

Where, X_{min} and X_{max} are the minimum and maximum values within the array of values. It is also worth mentioning that the normalization range should be selected based on the data points and the variation between them. If the data is large with large variation, and the selected range is {0-1} then it may not be easy to distinguish between the normalized values, as the difference between the normalized values would lie in least significant figures, say, 5 decimal points away, For example, the difference between 0.874824 and 0.874827 is hard to distinguish and not easily seen on a plot with wide range of values.

Application

This web service can be used in following applications:

- Comparing quantities with different units.
- Plotting various quantities with different units or wide range of values.
- Specific applications where normalized values are required.

4.4.3 Radar Chart

Radar chart, sometimes called the spider web chart, star chart, or polar chart, is a two dimensional multi-axis polar chart, used to display multivariable data of three or more than three quantities represented on axes. The axes are uniformly distributed equally spaced in 360 degrees. A sample radar chart showing budget allocation and actual spending among various sectors within an organization is shown in Figure 22.

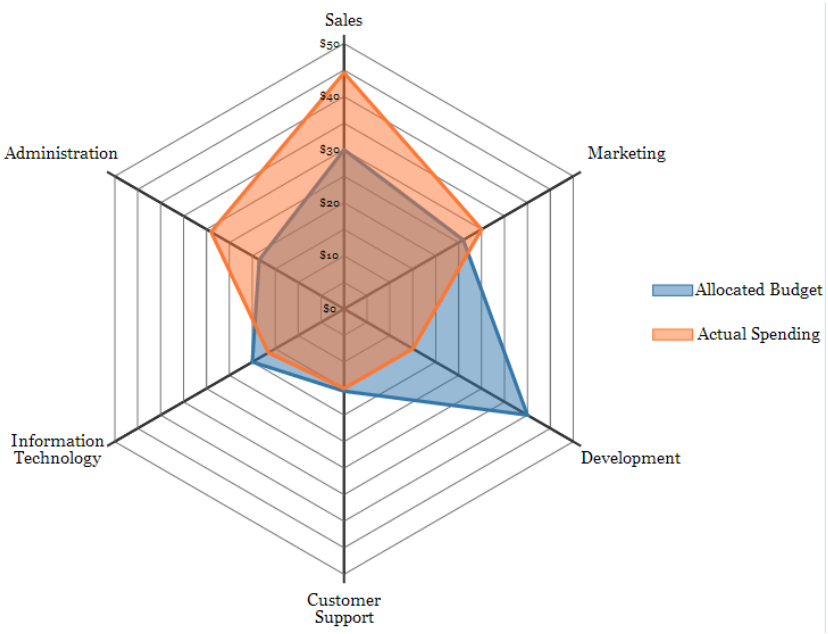

Figure 22: Sample Radar chart showing budget allocation and actual spending[20]

Functionality

In the radar chart function, made in MATLAB, each spoke represents one KPI. The lengths of all spokes are equal, and are normalized between maximum and minimum of the data, or optionally defined manually by the user. The data length of each spoke is proportional to the normalized value of each KPI. The legend of the radar chart represents the timestamps at which the data points are taken. For example: three data sets represent three values corresponding to each KPI having three timestamp in the legend. The title of the radar chart is defined by the names of the KPIs.

Application

The applications of radar chart vary depending upon the area of application. For example, in quality control, it is used to display performance variables of a system. In Energy analysis application, it may be used to display normalized values of different Energy KPIs at same timestamps.

4.4.4 Sankey Diagram

Sankey diagram is a type of flow diagram which represents the inputs and outputs of the system by ingoing and outgoing arrows. The width of the arrows is proportional to the flow quantity. It has many variations and visual representations, and could be open or closed loop. A sample open loop Sankey diagram showing various consumers of energy within a house along with the percent consumption is shown in Figure 23.

[20] Spider Chart, [WWW] [Accessed 07.04.2014] Available on: http://ie.microsoft.com/testdrive/Graphics/BusinessCharts/SpiderChart.xhtml

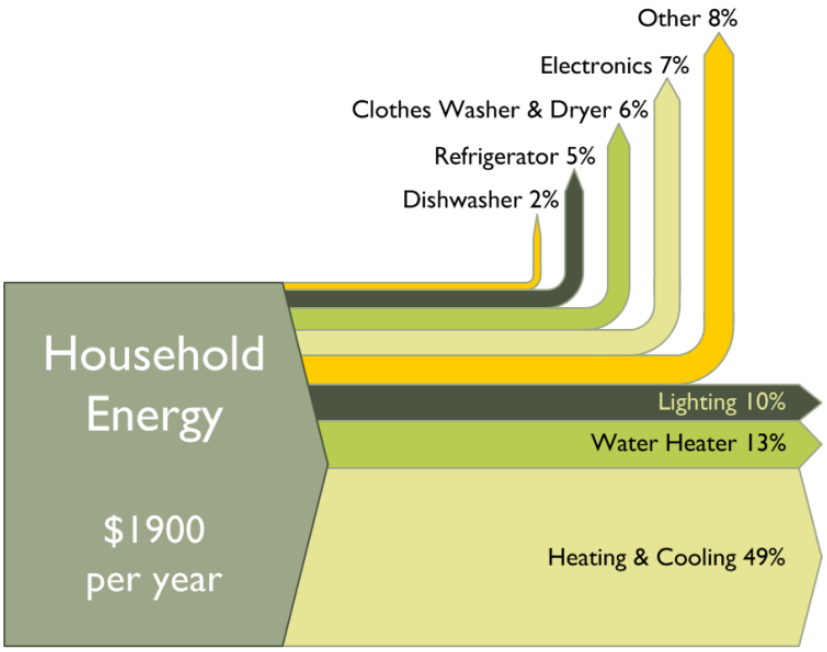

Figure 23: Sankey diagram showing consumers of energy in a typical house[21]

Functionality

In analysis module, Sankey diagram has been added using MATLAB and published as a web service. The input to the web services are the energy input, their Unit, the losses and the names of each input and output. The Sankey diagram is an open loop type, means there is no feedback energy source possible to the system.

Application

Sankey diagram is used to visualize inflow and outflow of resources (Energy, Money, material, etc.) between the processes or systems. In energy management systems, Sankey diagram is useful in visual representation of all the energy inputs to the process/system, the energy outputs and corresponding losses.

4.4.5 Data Correlation

Correlation defines the relationship between two variables or among more than two variables. The degree of correlation between any two data variables are given by correlation coefficient. Many linear and non-linear correlation coefficients have been introduced which vary in robustness. The most common of these is the Pearson correlation coefficient, which is sensitive only to a linear relationship between two data variables. Figure 24 shows how the correlation coefficients represent relations between various data sets. It can be seen that the positive and negative signs only represent the direction of slope.

[21] Sankey diagram, [WWW] [Accessed 07.04.2014] Available on: http://www.sankey-diagrams.com/wp-content/gallery/x_sankey_004/home-energy-use-sankey-diagram.png

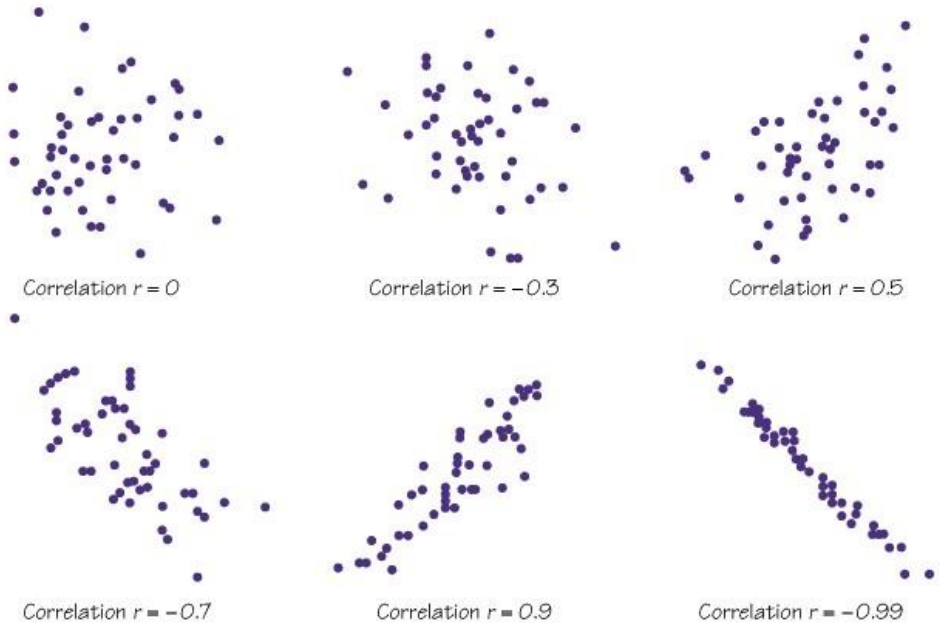

Figure 24: Correlation coefficients for various data sets[22]

Functionality

Pearson correlation coefficient varies between +1 and -1 for any two variables. positive sign denotes positive grade/slope of the linear fit, which means that by increasing value of one variable, the other also increases and vice versa, whereas negative sign denotes the negative grade/slope of the linear fit, which means that by increasing one variable the other variable decreases, and vice versa. Zero (0) correlation coefficient means that there is no dependency between the two variables and their behavior is 100% random between each other. Formula for correlation between two lists of values is given below [37].

$$R(i,j) = \frac{C(i,j)}{\sqrt{(C(i,i).C(j,j)}}$$

In the above formula, *C* is the covariance of input matrix, *i* represent the observation value (row), *j* represents variables (columns) and *R* represents the correlation coefficient matrix.

MATLAB has built-in function to calculate Pearson correlation coefficient of input matrix whose rows are observations and columns represents the variables. It returns a matrix of same size containing the correlation coefficients of variables among each other for corresponding observations.

This function is integrated in to java dynamic web project and published as web service. The user needs to provide at least two set of variables to find the correlation coefficient between them.

---

[22] Correlation graph, [WWW] [Accessed 07.04.2014] Available on: http://knottwiki.wikispaces.com/file/view/correlation_dot_graphs.jpg/136491277/correlation_dot_graphs.jpg

Application

Correlation can be computed between different KPIs or variables of interest. If more than two KPIs or Variables are given for correlation, the coefficients for all the possible combinations of the KPIs or variables are calculated. An example application of correlation is when it is desired to calculate how power consumption of certain robot increases with increasing number of pallets fed to the robot.

4.4.6 Pareto Chart

Pareto Chart, named after Vilfredo Pareto, is one of the seven basic tools of quality control. It is a chart containing both bars and line graph. The bars are placed in descending order of their values whereas the line graph represents the cumulative total starting from the bar with the highest value towards bar with lowest value. It has two vertical axes. Left vertical axes typically represents unit of measure, while the right vertical axes represents the cumulative percentage. The horizontal lower axis represents the names of the measures whose bars are made. A sample Pareto chart showing causes of late arrivals is shown in Figure 25.

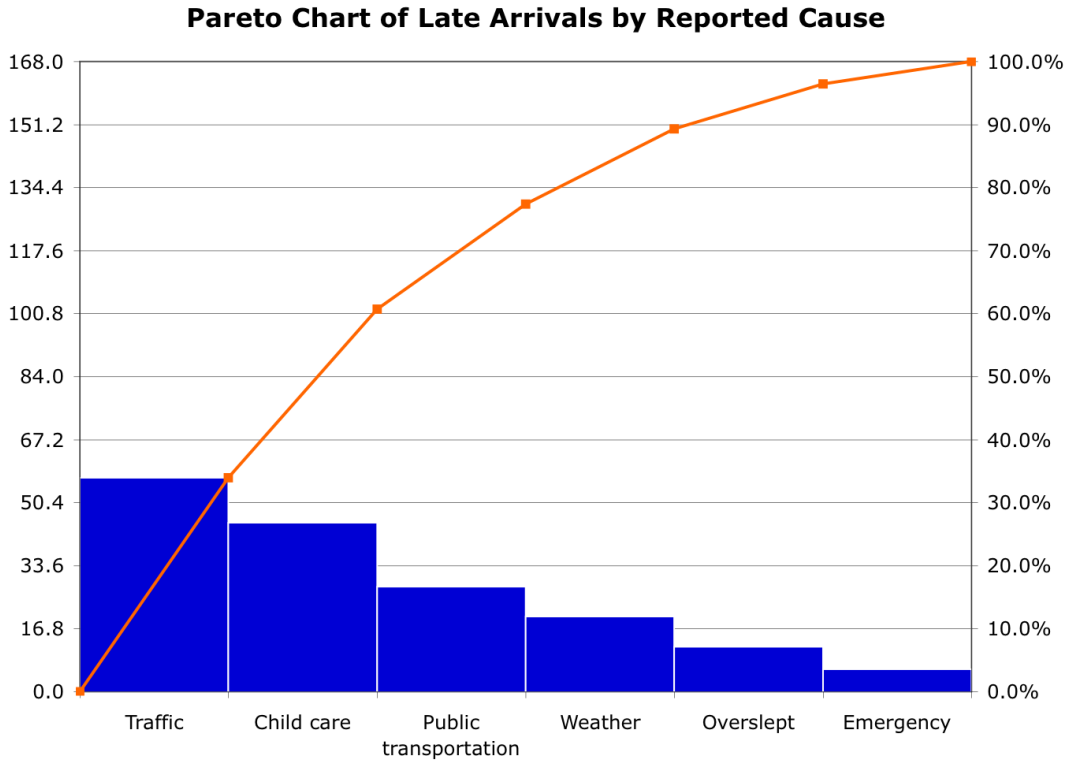

Figure 25: Sample Pareto Chart showing 100% cumulative causes[23]

Functionality

[23] Pareto Chart, [WWW] [Accessed 07.04.2014] Available on: http://upload.wikimedia.org/wikipedia/commons/8/8a/Pareto.PNG

MATLAB has a built-in function for making Pareto chart which displays 95% of the cumulative distribution. The function made for making Pareto Chart is deployed in Java and published as a web service. It takes the Names of the parameters, their values and the title for the Pareto Chart.

Application

Pareto chart is helpful in following ways:

- Determining the most important factors/areas/assets contributing towards a specific KPI or energy consumption measure. It gives indication to the factors which causes most of the effect.
- Analyzing data about the frequency of problems or causes in the process. Also, when there are many causes or problems and the most significant ones are to be identified.
- Summarize and display the relative importance of the differences between groups of data.

4.4.7 Shewhart Control Chart

Shewhart control chart, or simply control chart, is also one of the seven basic tools of quality control. It is named after Walter. A. Shewhart. In statistical process control, it is used to determine if a process is in state of statistical control. A sample control chart is shown in Figure 26.

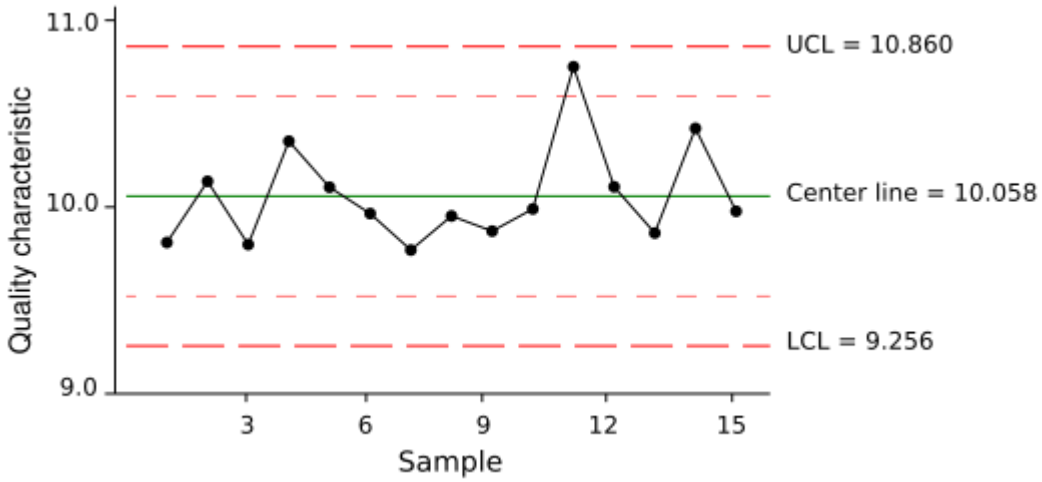

Figure 26: Sample Control Chart showing CL, LCL, and UCL

Functionality

MATLAB has built-in function to plot Shewhart Control Chart. The Chart plots the means of the subgroups (measurements containing replicate observations taken at same time) in time order. A Centre Line (CL) is drawn at the average of the means. Upper

and Lower Control Limits (UCL and LCL) are drawn at three standard deviations from the CL. Out of Control points are drawn as violation with red circle around the point.

Application

In manufacturing industry, some devices are programmed or dedicated to do processes cyclically, for example, a robot processing the incoming pallets with some predefined processes. The energy consumption by these repeated regular interval processes can be detected for any anomalies by the use of Shewhart Control Chart. If there is, for example, some irregularity in the process energy consumption of the robot, then the Control Chart will deviate from the CL, and if the deviation is larger than three standard deviations then a warning of violation is displayed in the chart.

4.5 Web service Objects

4.5.1 Service Request Object

Service Request Object is the generic schema of the XML according to which the data is sent to any of the available web services. A hierarchy diagram of the schema for the request object is shown in Figure 27.

As shown in Figure 27, the request object contains the parent element "*analyticServiceRequest*" which consists of two child elements "*serviceParameters*" and "*inputDataSeries*". *'serviceParameters'* contains all the parameters needed for the web service except the data itself, for example: Analytic method name, expected output format, Location information, etc. Each parameter is defined in a separate “metaData” element which consists of the key, the name of the parameter, and the value of the specific parameter. The "*inputDataSeries*" elements contain the actual data about certain variable. Each "*inputDataSeries*" is identified by a unique *'seriesID'*. "*inputDataSeries* further consists of "*metaData*" which contains the relevant information about the specific input variable, and the "*dataValue*" which contains the value of the variable along with its time stamp.

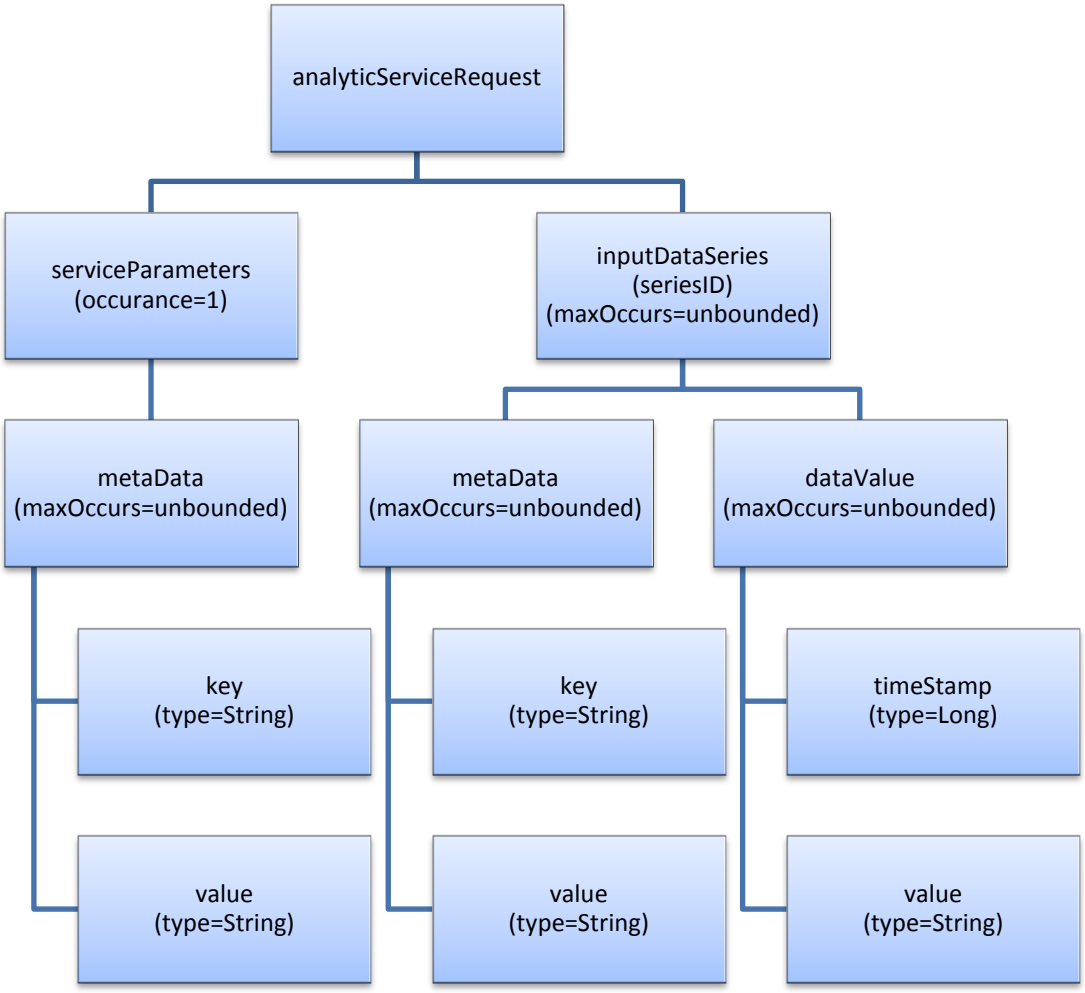

Figure 27: Hierarchy diagram of Service Request Object

A sample XML message representing service request object is shown below. The XSD of request object is given in appendix A.

```
<?xml version="1.0" encoding="UTF-8"?>
<analyticServiceRequest
xmlns="http://analyticservices.tut.fi/analyticService/request">
   <serviceParameters>
      <metadata key="analyticMethod">Radar Chart</metadata>
      <metadata key="numberOfKpi">3</metadata>
      <metadata key="expextedResultFormat">jpg</metadata>
      <metadata key="location">Lab2</metadata>
   </serviceParameters>
   <inputDataSeries seriesId="series001">
      <metadata key="dataType">staticKpi</metadata>
      <metadata key="seriesName">averagePowerConsumption</metadata>
      <metadata key="unit">kilowatt</metadata>
      <metadata key="inputType">main</metadata>
      <dataValue timestamp="1234567880">3.4</dataValue>
      <dataValue timestamp="1234567885">7.6</dataValue>
      </inputDataSeries>
   <inputDataSeries seriesId="series002">
      <metadata key="dataType">dynamicKpi</metadata>
      <metadata key="seriesName">totalPowerConsumption</metadata>
```

```
        <metadata key="unit">kilowatt</metadata>
        <metadata key="inputType">main</metadata>
        <dataValue timestamp="1234567880">7.6</dataValue>
        <dataValue timestamp="1234567885">4.2</dataValue>
    </inputDataSeries>
    <inputDataSeries seriesId="series003">
        <metadata key="dataType">staticKpi</metadata>
        <metadata key="seriesName">averagepowerPerDay</metadata>
        <metadata key="unit">kilowatt</metadata>
        <metadata key="inputType">main</metadata>
        <dataValue timestamp="1234567880">6.4</dataValue>
        <dataValue timestamp="1234567885">2.6</dataValue>
    </inputDataSeries>
</analyticServiceRequest>
```

4.5.2 Service Response Object

Service Response Object is the generic schema of the XML according to which the data is sent back by any web service as a response to the request. A hierarchy diagram of the schema for the response object is shown in Figure 28.

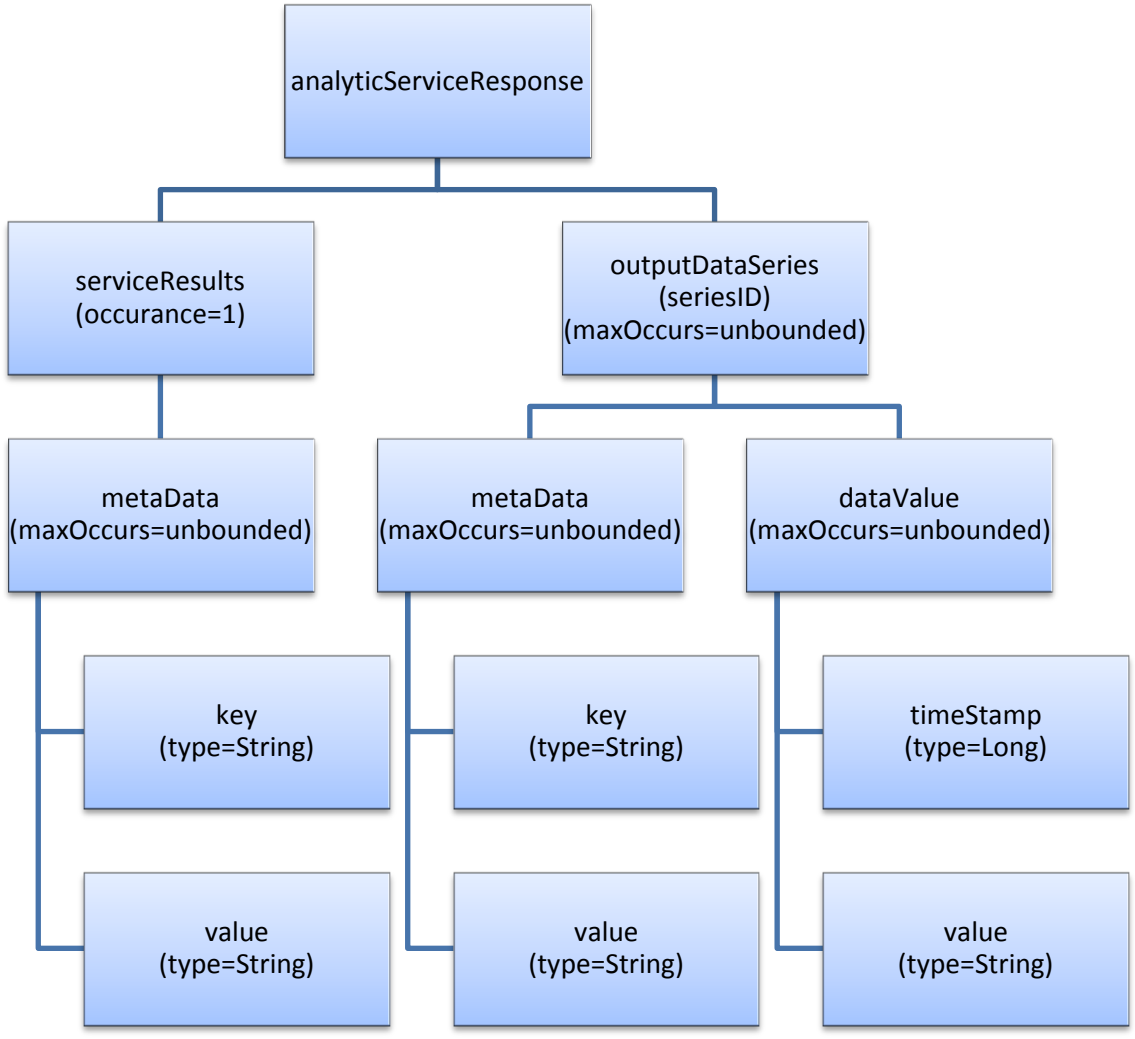

Figure 28: Hierarchy diagram of Service Response Object

As shown in Figure 28, the response object contains the parent element "*analyticServiceResponse*" which consists of two child elements "*serviceResults*" and "*outputDataSeries*". "*serviceResults*" element contains result of the web services any relevant information about the response. Each response parameter is defined in a separate "metaData" element which consist of the key, the name of the parameter, and the value of the specific parameter. The "*serviceResults*" might also contain a *metaData* element which contains figure returned by the web service. All the numerical results returned by web service are stored in the "*outputDataSeries*" element. Similar to "*inputDataSeries*", the "*outputDataSeries*" is also identified by a unique *'seriesID'*. The information about the numerical response data is stored in the "*metaData*" variables of the "*outputDataSeries*", while the actual numerical data is stored in the "*dataValue*" elements, having time stamp as the key and the actual value corresponding to the time stamp. A sample XML message representing service response object is shown below. The XSD of response object is given in appendix B.

```
<?xml version="1.0" encoding="UTF-8"?>
<analyticServiceResponse>
<serviceResults>
<metadata key="dataType">staticKpi</metadata>
<metadata key="dataSeriesId">series001</metadata>
<metadata key="seriesName">averagePowerConsumption</metadata>
<metadata key="location">Lab-4</metadata>
</serviceResults>
<outputDataSeries seriesId="series001">
<metadata key="dataType">staticKpi</metadata>
<metadata key="seriesName">averagePowerConsumption</metadata>
<metadata key="legendName">street1</metadata>
<metadata key="unit">kilowatt</metadata>
<dataValue timestamp="123123">50</dataValue>
<dataValue timestamp="132124">30</dataValue>
<dataValue timestamp="123125">25</dataValue>
<dataValue timestamp="132126">60</dataValue>
<dataValue timestamp="123127">55</dataValue>
</outputDataSeries>
<outputDataSeries seriesId="series003">
<metadata key="dataType">rawData</metadata>
<metadata key="seriesName">averagePowerConsumption</metadata>
<metadata key="legendName">street3</metadata>
<metadata key="unit">kilowatt</metadata>
<dataValue timestamp="123123">220</dataValue>
<dataValue timestamp="132124">350</dataValue>
<dataValue timestamp="123125">200</dataValue>
<dataValue timestamp="132126">260</dataValue>
<dataValue timestamp="123127">330</dataValue>
</outputDataSeries>
</analyticServiceResponse>
```

5. RESULTS

This chapter documents the results of the implemented analytic web services. The results are the outputs of each web service in response to the supplied data. Individual clients are made for testing of each web service. Data is provided through XML files containing the data values required by the web services. The Client sends the XML data to the web service in SOAP envelop and the web service returns the results to the client. The returned data and images are printed on the screen to better visualize the results. The evaluation criterion is unique for each web service and depends on the nature of the analytic service. Results of each analytic web service are shown in separate sections in this chapter. The data requirements of each web service are given in appendix D.

5.1 Data Consistency Check

Web service for filling missing values by interpolation was checked by sending various sets of data and obtained results using different methods of interpolations implemented in the web service. Because of large amount of data and better visual understanding, the original and interpolated data is shown graphically instead of tabular form.

For comparison of different interpolation methods, a sinusoidal data is sent to the web service with some missing values as shown in Figure 29. Results are obtained by interpolating data using different interpolation methods on the same data.

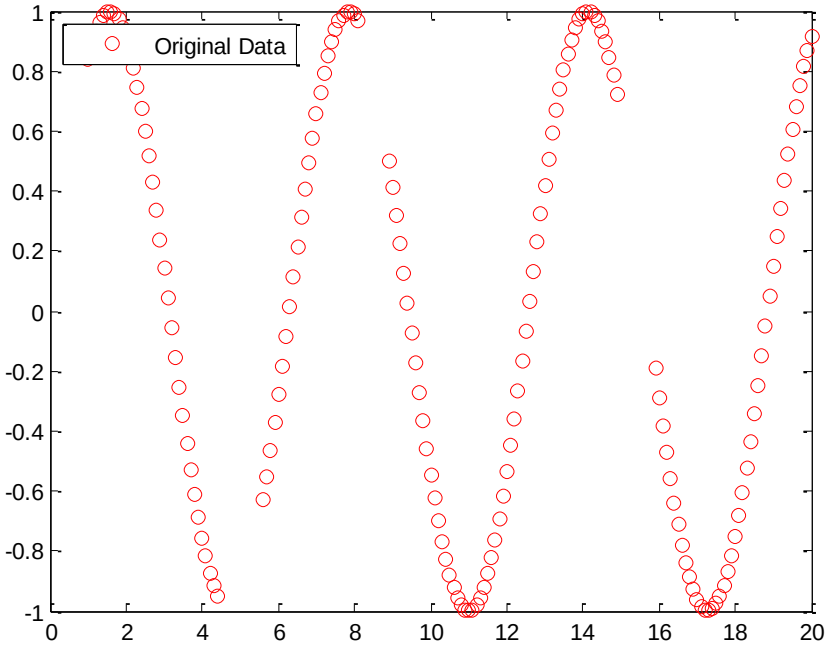

Figure 29: Original data with missing values

When the estimation method 0 (nearest neighbour interpolation) was selected, the result was not a very good approximation. The result is shown in Figure 30.

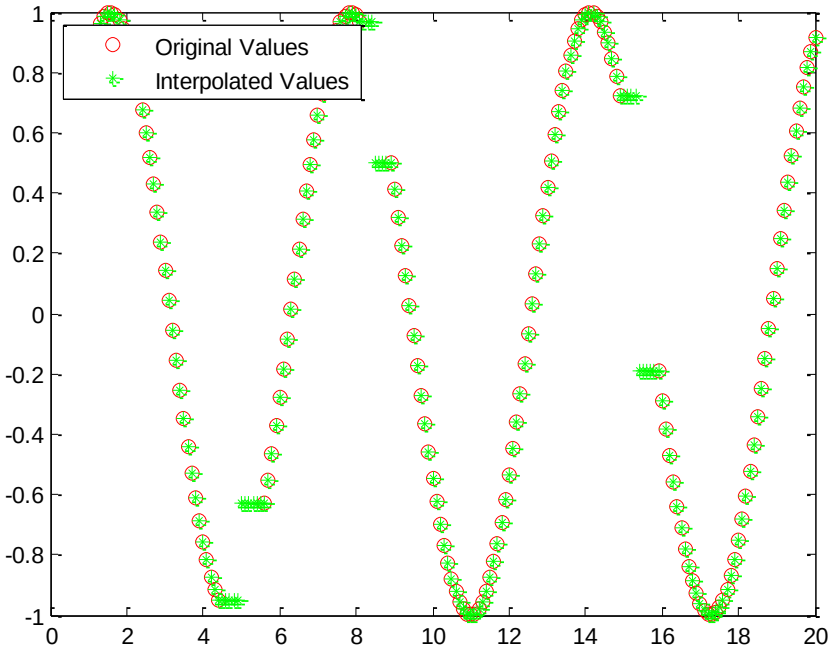

Figure 30: Data interpolation using nearest neighbour algorithm

It can be seen from the above figure that nearest neighbour algorithm only replicates the values of the nearest neighbours without any interpolation. The result obtained by estimating using linear interpolation is shown in Figure 31.

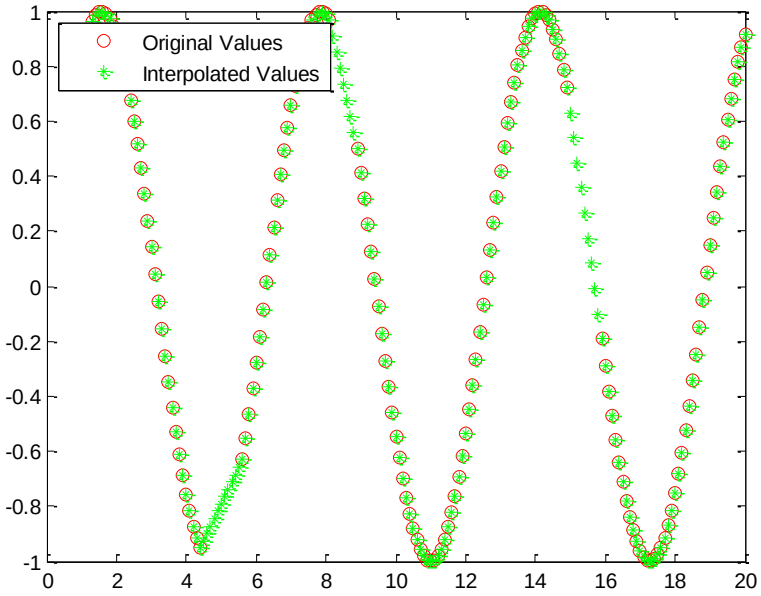

Figure 31: Data estimation using linear interpolation

It can be seen in Figure 31 that linear interpolation is better estimation of the missing values compared to nearest neighbour estimation method. The result obtained by using cubic interpolation is shown in Figure 32.

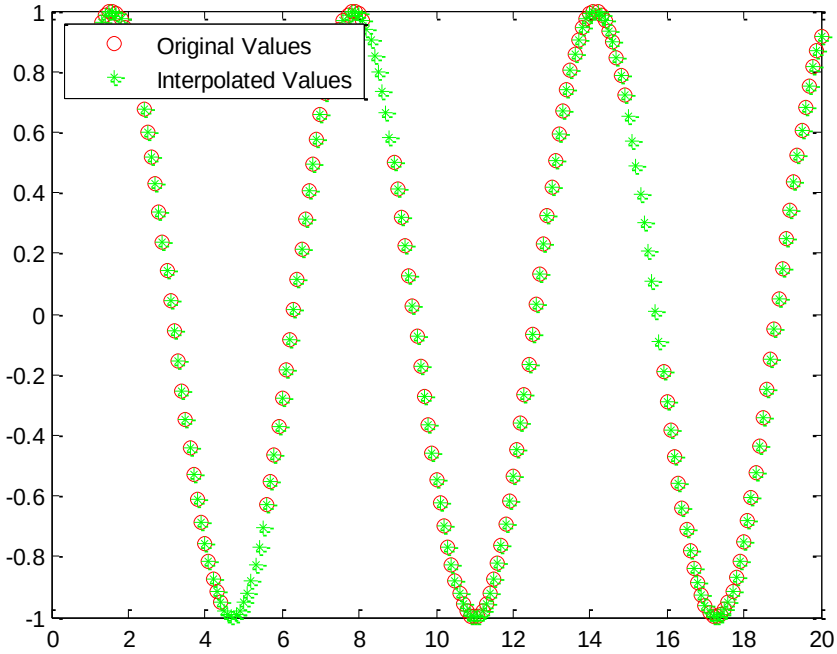

Figure 32: Data estimation using cubic interpolation

From the above figure it can be seen that cubic interpolation gives very good estimation of missing values. Based on these results, choice has to be made regarding which estimation method to use for missing data. The cubic interpolation has better results than linear and nearest neighbour estimation methods but on the other hand, the processing time and memory usage is high for cubic interpolation. It practically doesn't make much difference for fewer amounts of data, but these issues need to be considered when dealing with estimation of large amount of data.

5.2 Data Normalization

Normalization of data is tested by sending three sets of KPIs (Active electric power by cell, Root mean square voltage and peak load) to the web service, where each KPI has five values to be normalized as shown in Table 5.

Table 5: KPI values to be normalized

Active Electric Power by Cell	Root Mean Square Voltage	Peak Load
7	45	125
3	63	235
9	55	124
15	25	233
2	81	215
13	23	333
10	54	112
11	52	150

As a result, the normalized values are returned by the web service as shown in the Table 6. The normalization range was set between 1 and 10, so the maximum and minimum values in the list get values of 10 and 1 respectively.

Table 6: Normalized values of KPIs

Active Electric Power by Cell	Root Mean Square Voltage	Peak Load
4.4615	4.4138	1.5294
1.6923	7.2069	6.0090
5.8462	5.9655	1.4887
10.000	1.3103	5.9276
1.000	10.000	5.1946
8.6154	1.0000	10.000
6.5385	5.8103	1.0000
7.2308	5.5000	2.5475

5.3 Radar Chart

Five sets of KPIs were input to the web service of radar chart. Each KPI had three values (already normalized) corresponding to three different time stamps as shown in Table 7.

Table 7: Normalized KPI values for radar chart

Time stamps	Location: FASTory				
	Process units by cell	Cell production rate	Energy consumption per product	Average unit production time	Power consumption per cell
1390329383000	5	3	7	9	5
1390284923000	7	1	4	3	9
1390298128000	3	9	2	7	8

The resulting figure obtained from the web service is shown in Figure 33. It can be seen that the time stamps of each value are represented in legend of the chart with corresponding color coding with respect to each value. The axes of the radar chart are automatically adjusted corresponding to the maximum and minimum values of the KPIs and they can also be manually adjusted by the user.

Although the units for each KPI are different from each other, still it can be plotted together using radar chart with their normalized values. This helps in comparison of the KPIs with normalized values. The location name can also be shown in title of the graph.

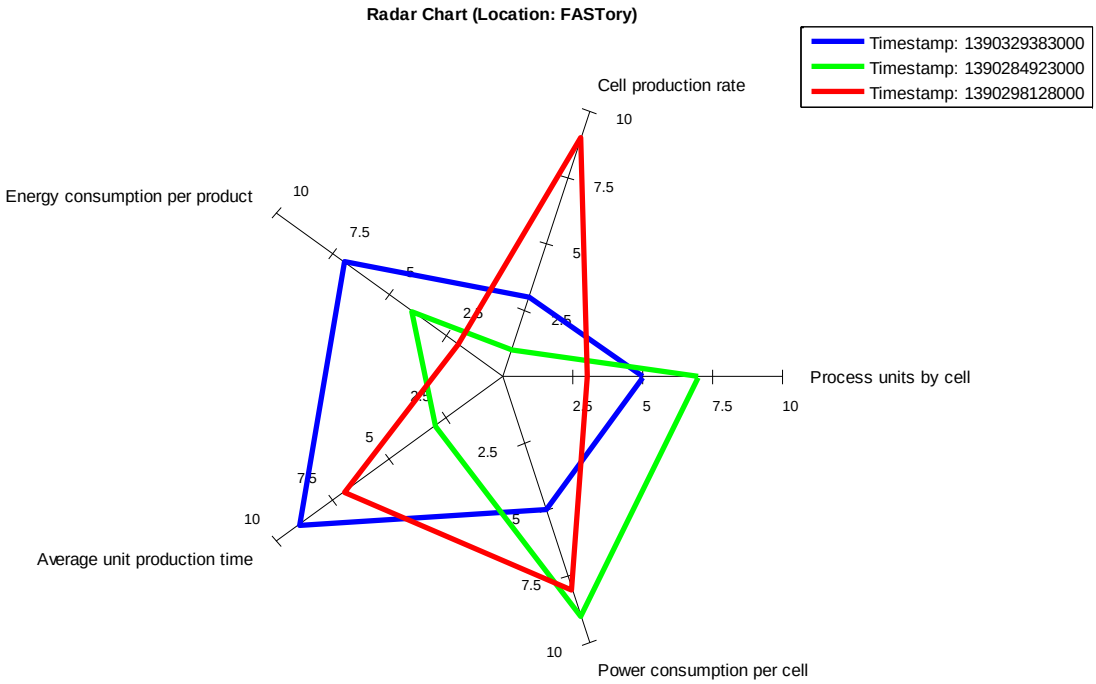

Figure 33: Radar Chart showing values of five KPIs

In the above figure, information can be extracted by comparing any two or more than two KPIs for specific time stamps. For example: comparing process units by cell and average unit production time.

5.4 Sankey Diagram

Sankey diagram is simple yet helpful diagram to show the energy input and output across an asset, device or process. A message containing following information is sent to the web service.

Table 8: Inputs to the Sankey diagram web service

Name	Type	Value
Electricity	Input	75
Heat	Input	32
Conveyors	Loss	10
Controllers	Loss	5
Routers	Loss	2.8
Robots	Loss	89
	Unit	Watts

The web service returns the Sankey Diagram as shown in Figure 34.

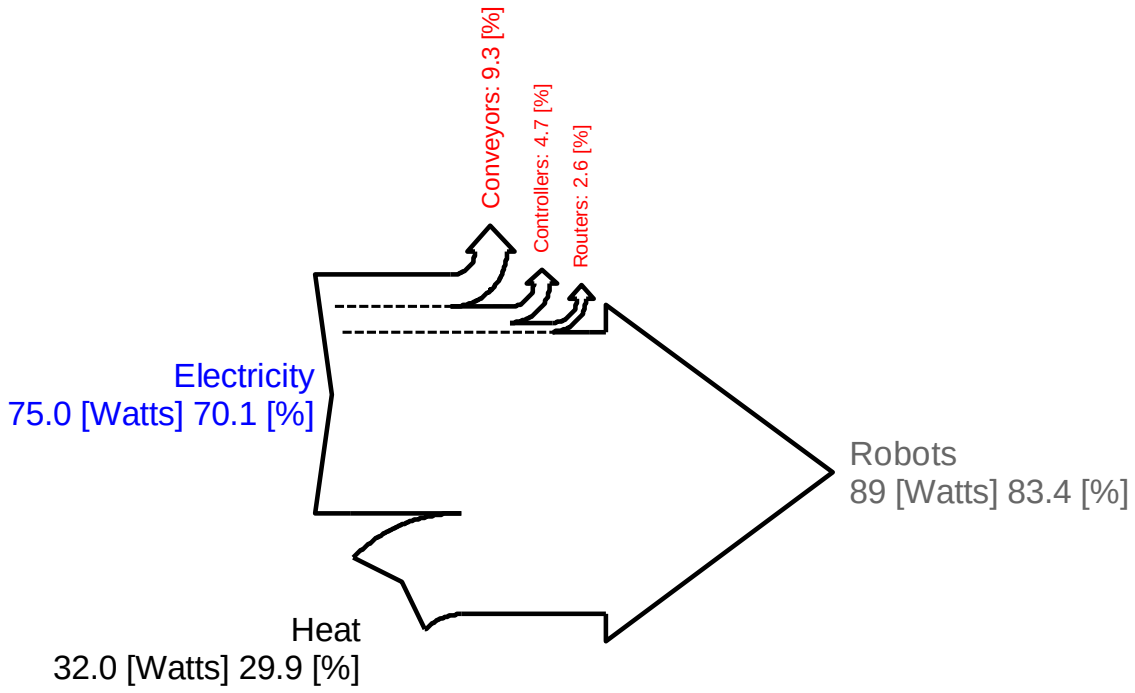

Figure 34: Sankey diagram returned from web service

The above figure shows the utilization of energy (heat and electricity) in the production line. The main consumers are the robots processing the pallets consuming 89 Watts which constitutes 83.4% of the total energy consumption. The conveyors, controllers and routers consume 9.3%, 4.7% and 2.6% of the total input energy, respectively. These results conclude that robots are one of the highest consumers of energy in the manufacturing facility. In-depth analysis of the process operations of robots should be done to identify the possibility of reduction in energy consumptions of the robots.

5.5 Data Correlation

In order to see the correlation between different KPIs, four series of KPI data was provided to the web service to compute the coefficients of Pearson's Correlation as shown in Table 9.

The correlation coefficients measure the degree of correlation between the KPIs, i.e. how the change in one KPI is affecting another set of KPI. If there is no correlation between the two KPIs, then the value of the coefficient is zero. Plus and minus signs indicate the positive and negative correlation between the KPIs.

Table 9: Values of four lists of KPIs

Location: FASTory			
Power consumption per cell	Peak load	Active electric energy consumption by cell	Units produced by line
2	7	9	77
7	8	8	66
4	9	7	55
2	9	7	55
6	9	6	44
9	10	6	44
8	11	5	33
9	11	5	22
50	12	5	11
11	12	4	4
120	13	2	2

The web service computed all the coefficients and stored the result in the response message. Snap shot of the result is shown in Figure 35.

```
Invoking correlate...
correlate.result=fi.tut.analyticservices.analyticservice.response.AnalyticServiceResponse@2f5ba4b
CorrCoef between Power consumption per cell and Power consumption per cell = 1.0
CorrCoef between Power consumption per cell and Peak load = 0.6720882087012765
CorrCoef between Power consumption per cell and Active Electricity consumption by cell = -0.7274475232489725
CorrCoef between Power consumption per cell and Units produced by line = -0.6286326800452496
CorrCoef between Peak load and Power consumption per cell = 0.6720882087012764
CorrCoef between Peak load and Peak load = 1.0
CorrCoef between Peak load and Active Electricity consumption by cell = -0.9605787860802001
CorrCoef between Peak load and Units produced by line = -0.9795665777842144
CorrCoef between Active Electricity consumption by cell and Power consumption per cell = -0.7274475232489725
CorrCoef between Active Electricity consumption by cell and Peak load = -0.9605787860802001
CorrCoef between Active Electricity consumption by cell and Active Electricity consumption by cell = 1.0
CorrCoef between Active Electricity consumption by cell and Units produced by line = 0.9485881738154931
CorrCoef between Units produced by line and Power consumption per cell = -0.6286326800452496
CorrCoef between Units produced by line and Peak load = -0.9795665777842145
CorrCoef between Units produced by line and Active Electricity consumption by cell = 0.948588173815493
CorrCoef between Units produced by line and Units produced by line = 1.0
```

Figure 35: Correlation Coefficients among four list of KPIs

It can see from the above figure that the correlation coefficient of any series of KPI with itself is always 1.0 which obvious. The correlation coefficient between two different series of data can be any value between -1 and 1.

From Figure 35, it can conclude that the KPI-1 (Power consumption per cell) has negative correlation with KPI-3 (Active electric energy consumption by cell) with coefficient value of -0.72 approx. This means that when KPI-1 is increasing, the KPI-3 is decreasing. Similarly, correlation of KPI-1 with KPI-2 (Peak load) is positive while with KPI-4 (Units produced by line) is negative. The result also shows high positive correlation between KPI-3 and KPI-4 with coefficient value of approx. 0.948.

5.6 Pareto Chart

Pareto Chart WS was invoked by providing the values of KPI (Power consumption per hour) values for six consumers of manufacturing facility, as shown in Table 10.

Table 10: Power Consumption values of consumers in FASTory

S.No	Consumer	Power Consumption
1	Controllers	15
2	Conveyors	90
3	Lights	100
4	Robots	400
5	Routers	2
6	HVAC	50

As a result of the above values, WS returned a Pareto Chart as shown in Figure 36. It can be seen that the consumer names are sequentially placed in descending order of their values. Robots being the highest consumer gets in first place, followed by Lights, Conveyors and HVAC.

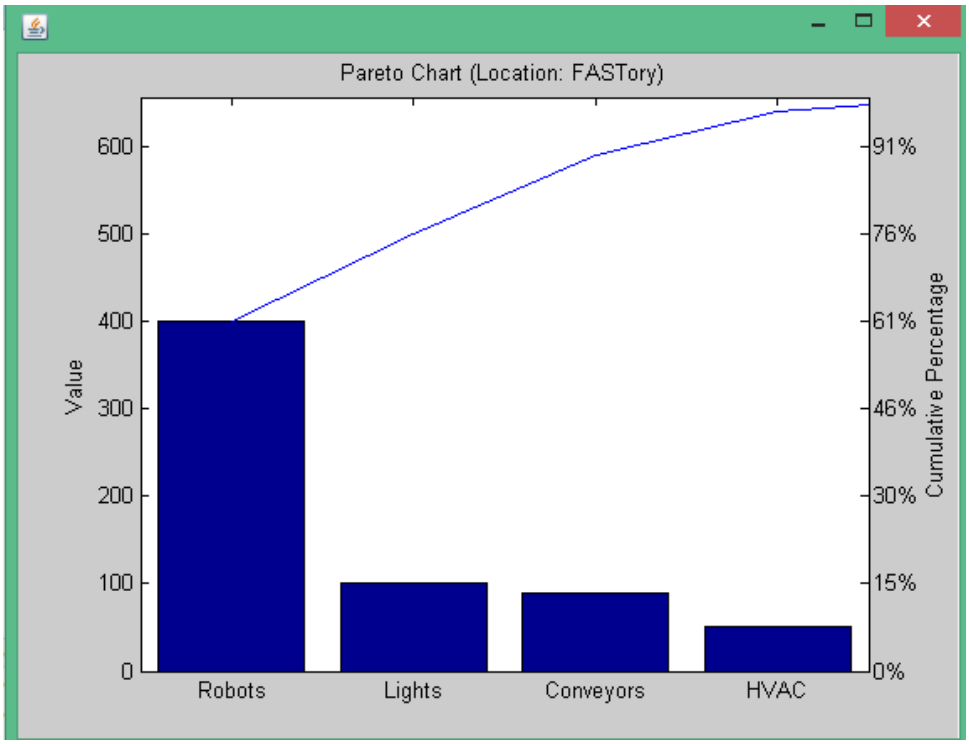

Figure 36: Pareto Chart showing highest power consumptions

It is to be noted that Controllers and Routers are not included in the above chart. It is due to the reason that this Pareto Chart is set to show 95% of the cumulative effect. It helps in showing only the major components responsible for the maximum consumption. The right vertical axis shows the value of power consumption while cumulative percentage can be seen on the left side vertical axis. The thin blue line represents the

cumulative percentage at each bar (different consumers in this case). In this specific case, this chart gives us information regarding highest power consumptions of various consumers. This chart can also be used in calculating different KPIs of various assets and processes- in the manufacturing facility.

5.7 Shewhart Control Chart

The analytic web service for Shewhart Control Chart was invoked by supplying data of specific process executed by Robot. Four set of measurements are recorded as shown in Table 11.

Table 11: Power Consumption of Robot

Power Consumption: Robot Processing			
Measurement-1	Measurement-2	Measurement-3	Measurement-4
75	74	79	75
12	18	14	22
43	43	53	46
41	41	48	51
14	44	64	74
21	31	51	81
12	14	14	22
32	36	37	32
43	47	41	23
64	34	64	66
10	50	20	18

As a response to the data above, Figure 37 is returned by the web service. It can be seen that the first, second and seventh readings are violating the three standard deviations boundary. These violations give essential information about deviation of process variables under consideration and should be checked. For normal operation of the robot, the data points should be near to centre line, which is average of the means of all measurements.

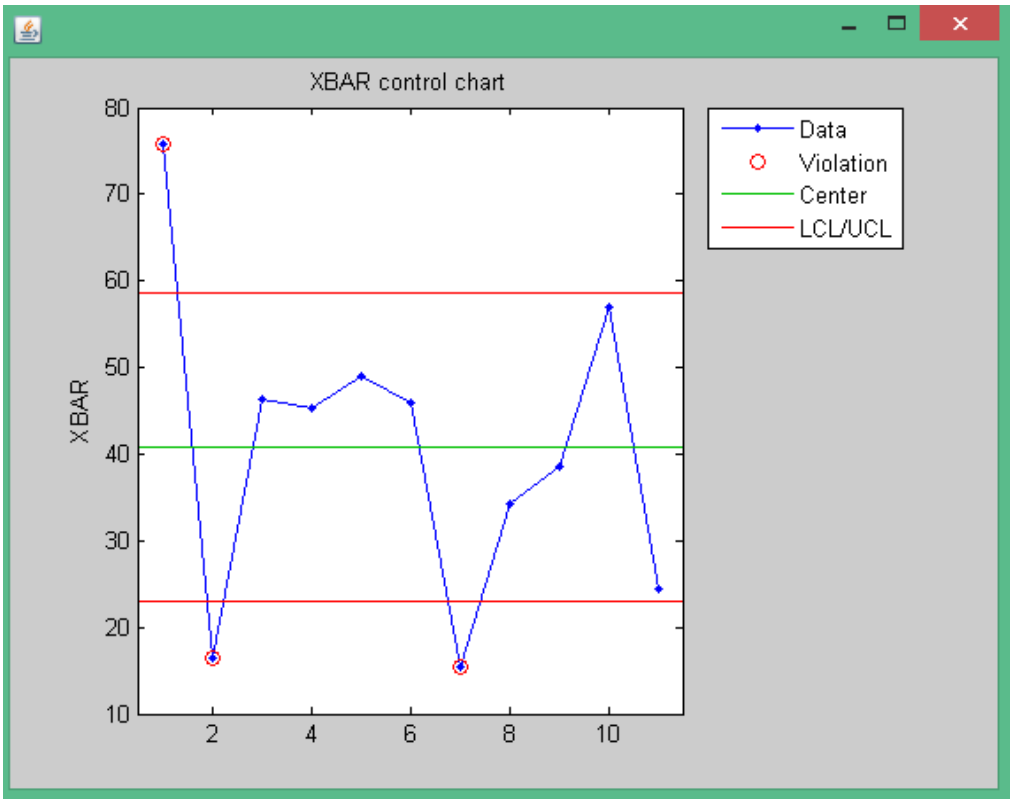

Figure 37: Shewhart Control Chart for given measurements of Robot

5.8 Recommendations for use of analytic services

Based on the results of the analytic web services the following recommendations are proposed for the use of these services.

Checking for missing or null values in the data can be performed using '*data check*' web service. This web service ensures that the data has no missing or null values. This web service is recommended to invoke before doing any other operation on the data. The '*normalize data*' service normalizes the data between ranges of values. Normalization makes the data scale-invariant and hence multiple KPIs with different units can be plotted together and compared in a single chart. The '*radar chart*' service enables plotting of different KPIs on single chart. Radar chart is multi-axis chart, so more than two KPIs can be compared among each other. The '*Pareto chart*' service shows the consumers of energy or other energy related parameter in descending order. The 95% of cumulative effect is shown, so the Pareto chart filters out the assets which consume less energy. The '*Sankey diagram*' service visualizes the flow of energy across an asset or process in the manufacturing system. It shows the energy inputs and losses in assets or processes. Through Sankey diagram it is very easy to see which assets or processes consumes most of the energy. The '*data correlation*' service correlates two or more than two sets of KPI data and finds the correlation coefficient among the sets of data. The

correlation coefficient gives the relation between any two sets of KPI data and how the change in one KPI can bring changes in another KPI. The last implemented web service is the '*control chart*' which shows if the process is in state of statistical control. It detects anomalies in the process data and issues warning when the deviation in the data is more than three standard deviations from the average of the mean values of the measurements. This can be used to check if an asset or process has any anomaly in energy consumption.

The above set of analytic web services can be used in conjunction to calculate energy footprint of manufacturing assets. It helps in identifying significant energy aspects which consumes most of the energy or have highest potential of saving energy.

6. CONCLUSIONS

This chapter contains conclusions based on the results of the implemented analytic web services. Section 6.1 contains the contribution of the thesis work, followed by conclusions and discussion about the implemented analytic web services in section 6.2. Recommendations for future work are given in section 6.3. Recommendations for optimizing performance of the web services are given in section 6.4.

6.1 Contribution

Analytic tools have been developed in the research work of this thesis. The analytic tools implement analysis algorithms to calculate energy footprint of manufacturing system. The developed analytic tools are published as web services implementing service-oriented architecture. This enables modularity and expansion of web services to obtain required information from the services. XML message schemas for request and response messages of the web services are also proposed.

6.2 Conclusion and Discussion

Analytic tools have been developed, in this thesis work, to analyze energy data for calculating energy footprint of manufacturing assets. The developed tools were demonstrated by publishing and invoking them as web services. Then the analytic tools were demonstrated to calculate the energy footprint of manufacturing assets by providing required data to the services.

The implementation of the web services is performed in Java. Because Java is object-oriented and also platform independent, it adds advantage to the implementation. Integrating functionalities of MATLAB in java gives additional advantage of efficient and effective data analysis. The functions of MATLAB compiled in to Java classes are encrypted, so that the functions can be redistributed among users without giving the actual code.

6.3 Future work

In the implemented methodology, the web services return static data to the clients. This can be improved by using 3D interactive figures instead of static images. This will enable the client to interact with the data and manipulate it to get desired information.

Control over web services for data analysis can be accomplished by implementing analysis module manager. In the implemented methodology of this thesis work, analysis module manager is not integrated and the services are invoked independently by the client. However, for better orchestration of web services a separate module manager can be utilized.

6.4 Performance Optimization

The MATLAB deployed applications in Java uses the MATLAB Compiler Runtime (MCR) at runtime. The MCR is loaded when class in the deployed component is instantiated for the first time. The loading of MCR usually takes time between 20 seconds to 1 minute which adds additional delay in the execution of the program. Performance of the web services can be improved by loading MCR at server startup instead of loading it during first initialization of the deployed component by user.

Incoming message is parsed in the implementation class of the web services in Java. Message parsing can be improved by making separate parser classes or implement within MATLAB functions. Similar improvements can be brought in copying data to outgoing message as well.

REFERENCES

[1] U.S Energy information administration [WWW]. [Accessed on 27.03.2014]. Available at: http://www.eia.gov/tools/faqs/faq.cfm?id=447&t=3.

[2] ABB Industrial Energy Efficiency Review 2013, [WWW] [Accessed 26.03.2014], Available on: http://www.economistinsights.com/sustainability-resources/analysis/intelligent-manufacturing-targeting-better-energy-efficiency

[3] Tracking Industrial Energy Efficiency and CO2 emissions, International Energy Agency, [WWW] [Accessed on : 07.04.2014] Available on: https://www.iea.org/publications/freepublications/publication/tracking_emissions.pdf

[4] Official Statistics of Finland (TilastoKeskus), [WWW] [Accessed 24.03.2014] Available on: http://tilastokeskus.fi/til/tene/2012/tene_2012_2013-10-31_en.pdf

[5] Park C.W., Kwon K.S., Kim W.B., Min B.K., Park S.J., Sung I.H., Yoon Y.S., Lee K.S., Lee J.H., Seok J., "Energy Consumption Reduction Technology in Manufacturing – A Selective Review of Policies, Standards, and Research", *International Journal of Precision Engineering and Manufacturing*, Vol. 10, No. 5, pp. 151-173

[6] Commission of the European Communities, "A European Strategic Energy Technology Plan (SET-PLAN): Towards a low carbon future," COM 723, 2007.

[7] Lei Yang; Deuse, J.; Droste, M., "Energy efficiency at energy intensive factory - a facility planning approach," *IEEE 18th International Conference on Industrial Engineering and Engineering Management* (IE&EM), vol. Part 1, pp.699,703, 3-5 Sept. 2011

[8] Bladh, I., "Energy Efficiency in Manufacturing," European Commission, 2009.

[9] Dietmair, A. and Verl, A., "Energy Consumption Forecasting and Optimization for Tool Machines," *Modern Machinery Science Journal*, pp. 62-67, 2009.

[10] Jayachandran R., Singh S., Goodyer J, Popplewell K.: "The design of a sustainable manufacturing system: A case study of its importance to product variety manufacturing". *In proceedings of Innovative Production Machines and Systems Conference* (IPROMS), 2006

[11] Florea, A.; Montemayor, J.A.G.I.; Postelnicu, C.; Lastra, J.L.M., "A cross-layer approach to energy management in manufacturing," *10th IEEE International Conference on Industrial Informatics* (INDIN), 2012, pp.304,308, 25-27 July 2012

[12] I.S. EN 16001:2009. Energy Management System Implementation guideline- Requirements with guidance of use, Sustainable Energy Ireland, 32 p.

[13] Energy Office, [WWW] [Accessed on 01.04.2014], Available at: http://www.energyoffice.org/english/tools/emanagement/definition/main.html

[14] Luyang Zhang; Jiaqi Li; Ming Yu, "An Integration Research on Service-oriented Architecture (SOA) for Logistics Information System", *IEEE International Conference on* Service *Operations and Logistics, and Informatics, 2006. SOLI '06.*, pp.1059,1063, 21-23 June 2006

[15] Stoffels, P.; Boussahel, W.M.; Vielhaber, M.; Frey, G., "Energy engineering in the virtual factory," *IEEE 18th Conference on Emerging Technologies & Factory Automation* (ETFA), 2013, pp.1,6, 10-13 Sept. 2013

[16] Herrmann, C. & Thiede, S. 2009, "Process chain simulation to foster energy efficiency in manufacturing", CIRP *Journal of Manufacturing Science and Technology*, vol. 1, no. 4, pp. 221-229.

[17] Endo, M.; Nakajima, H.; Hata, Y., "Simplified Factory Energy Management System based on operational condition estimation by sensor data," *IEEE International Conference on Automation Science and Engineering* (CASE), pp.14,19, 20-24 Aug. 2012

[18] Energy Management Systems, [WWW] [Accessed 02.04.2014] Available at: https://www.honeywellprocess.com/en-US/explore/products/energy-and-emissions-management/Pages/energy-management-systems.aspx

[19] Fujitsu Official Website, [WWW] [Accessed on : 07.04.2014], Available at: http://www.fujitsu.com/global/about/environment/factories/highlight2013-1.html

[20] ISO 50001:2011, Energy Management Systems - Requirements with guidance of use, 22 p.

[21] I.S. EN 16001:2009. Energy Management System Technical guideline- Requirements with guidance of use, Sustainable Energy Ireland, 28 p.

[22] Cannata, A.; Karnouskos, S.; Taisch, M., "Energy efficiency driven process analysis and optimization in discrete manufacturing," *35th Annual Conference of IEEE on Industrial Electronics, 2009. IECON '09*, pp.4449,4454, 3-5 Nov. 2009

[23] Tomasz Markiewics, "Using MATLAB software with Tomcat server and Java platform for remote image analysis in Pathology", Diagn Pathol. 2011 Mar 30;6 Suppl 1:S18. doi: 10.1186/1746-1596-6-S1-S18.

[24] Huhns, M.N.; Singh, M.P., "Service-oriented computing: key concepts and principles," *Internet Computing, IEEE* , vol.9, no.1, pp.75,81, Jan-Feb 2005.

[25] Wang, H., Huang, J.Z., Qu, Y., Xie, J. “Web services: Problems and Future Directions”, 2004, *Journal of Web Semantics*, vol. 1, pp. 309-320.

[26] Web Service Architecture, [WWW] [Accessed 03.04.2014] Available on: http://www.w3.org/TR/ws-arch/

[27] Web Services tutorial, [WWW] [Accessed 03.04.2014] Available on: http://docs.oracle.com/javaee/6/tutorial/doc/gijvh.html

[28] Reusable Service, [WWW] [Accessed 18.04.2014], Available on: http://www.oracle.com/technetwork/articles/bechara-reusable-service-087796.html

[29] Baghdadi, Y., "A framework to select an approach for Web services and SOA development," *International Conference on Innovations in Information Technology (IIT), 2012*, pp.277,282, 18-20 March 2012

[30] Colombo A.W. and Karnouskos S., "Towards the Factory of the Future: A Service-Oriented Cross-Layer Infrastructure," in *European Telecommunications Standards Institute* (ETSI) Book on: "ICT Shaping the World - A Scientific View", chapter 6: John Wiley and Sons, 2009.

[31] Karnouskos, S., Colombo, A.W., Lastra, J.L.M. and Popescu, C. 2009, "Towards the energy efficient future factory", *7th IEEE International Conference on Industrial Informatics*, 2009. INDIN 2009., pp. 367.

[32] Waris, M.; Khan, S.A.; Fakhar, M.Z., "Factors effecting service oriented architecture implementation," *Science and Information Conference (SAI), 2013*, pp.1,8, 7-9 Oct. 2013

[33] Mark H., Eibe F., Geoffrey H., Bernhard P., Peter R., Ian H. W. (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.

[34] Hussain, S.; Gabbar, H.A.; Musharavati, F.; Pokharel, S., "Key performance indicators (KPIs) for evaluation of energy conservation in buildings," *IEEE International Conference on Smart Energy Grid Engineering* (SEGE), pp.1,6, 28-30 Aug. 2013

[35] Hossain, M.M., Portlet based presentation of energy KPIs in SOA-enabled manufacturing facilities targeting holistic energy management, MSc Thesis, Tampere 2014, Tampere University of Technology, Department of Mechanical Engineering and Industrial Systems, 95 p.

[36] Thomas, J.P.; Thomas, M.; Ghinea, G., "Modeling of Web services flow," *IEEE International Conference on E-Commerce, 2003. CEC 2003*, pp.391,398, 24-27 June 2003.

[37] Correlation Coefficients, [WWW] [Accessed 07.04.2014] Available on: http://www.mathworks.se/help/matlab/ref/corrcoef.html

APPENDIX A: XML SCHEMA OF SERVICE REQUEST MESSAGE

```
   <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" element-
FormDefault="qualified" attributeFormDefault="unqualified">

   <!-- Definition of analyticServiceRequest element, the base element of the
schema -->
   <xs:element name="analyticServiceRequest">
      <xs:annotation>
            <xs:documentation>analyticServiceRequest contains the data and
parameters required for analytic service execution</xs:documentation>
      </xs:annotation>
      <xs:complexType>
            <xs:sequence>
                  <xs:element ref="serviceParameters" minOccurs="1" max-
Occurs="1"/>
                  <xs:element ref="inputDataSeries" maxOccurs="unbounded"/>
            </xs:sequence>
      </xs:complexType>
   </xs:element>

   <!-- Definition of serviceParameters element -->
   <xs:element name="serviceParameters">
      <xs:annotation>
            <xs:documentation>serviceParameters element contains all the pa-
rameters to be passed to the analytic service</xs:documentation>
      </xs:annotation>
      <xs:complexType>
            <xs:sequence>
                  <!-- metadata elements to describe the serviceParameters -
->
                  <xs:element ref="metadata" maxOccurs="unbounded"/>
            </xs:sequence>
      </xs:complexType>
   </xs:element>

   <!-- Definition of inputDataSeries element -->
   <xs:element name="inputDataSeries">
      <xs:annotation>
            <xs:documentation>inputDataSeries contains metadata and concrete
data of a certain data entity</xs:documentation>
      </xs:annotation>
      <xs:complexType>
            <xs:sequence>
                  <!-- metadata elements to describe the inputDataSeries -->
                  <xs:element ref="metadata" maxOccurs="unbounded"/>
                  <!-- dataValues of the inputDataSeries -->
                  <xs:element ref="dataValue" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="seriesId" type="xs:string"></xs:attribute>
      </xs:complexType>
   </xs:element>

   <!-- Definition of metadata element -->
   <xs:element name="metadata">
      <xs:annotation>
            <xs:documentation>metadata is used to store key-value pairs that
describe service parameters and data series</xs:documentation>
      </xs:annotation>
      <xs:complexType>
            <xs:simpleContent>
                  <xs:extension base="xs:string">
                        <xs:attribute name="key" type="xs:string"
use="required"/>
                  </xs:extension>
            </xs:simpleContent>
      </xs:complexType>
```

```
    </xs:element>

    <!-- Definition of dataValue element -->
    <xs:element name="dataValue">
        <xs:annotation>
                <xs:documentation>A set of data value elements is used to store
data values with their corresponding timestamps</xs:documentation>
        </xs:annotation>
        <xs:complexType>
                <xs:simpleContent>
                        <xs:extension base="xs:string">
                                <xs:attribute name="timestamp" type="xs:long"
use="required"/>
                        </xs:extension>
                </xs:simpleContent>
        </xs:complexType>
    </xs:element>

    </xs:schema>
```

APPENDIX B: XML SCHEMA OF SERVICE RESPONSE MESSAGE

```
   <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" element-
FormDefault="qualified" attributeFormDefault="unqualified">

   <!-- Definition of analyticServiceResponse element, the base element of the
schema -->
   <xs:element name="analyticServiceResponse">
      <xs:annotation>
            <xs:documentation>analyticServiceResponse contains the data and
parameters received from analytic service execution</xs:documentation>
      </xs:annotation>
      <xs:complexType>
            <xs:sequence>
                  <xs:element ref="serviceResults" minOccurs="1" max-
Occurs="1"/>
                  <xs:element ref="outputDataSeries" maxOccurs="unbounded"/>
            </xs:sequence>
      </xs:complexType>
   </xs:element>

   <!-- Definition of serviceResults element -->
   <xs:element name="serviceResults">
      <xs:annotation>
            <xs:documentation>serviceResults element contains all the parame-
ters to be received from the analytic service</xs:documentation>
      </xs:annotation>
      <xs:complexType>
            <xs:sequence>
                  <!-- metadata elements to describe the serviceResults -->
                  <xs:element ref="metadata" maxOccurs="unbounded"/>
            </xs:sequence>
      </xs:complexType>
   </xs:element>

   <!-- Definition of outputDataSeries element -->
   <xs:element name="outputDataSeries">
      <xs:annotation>
            <xs:documentation>outputDataSeries contains metadata and concrete
data of a certain data entity</xs:documentation>
      </xs:annotation>
      <xs:complexType>
            <xs:sequence>
                  <!-- metadata elements to describe the inputDataSeries -->
                  <xs:element ref="metadata" maxOccurs="unbounded"/>
                  <!-- dataValues of the inputDataSeries -->
                  <xs:element ref="dataValue" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="seriesId" type="xs:string"></xs:attribute>
      </xs:complexType>
   </xs:element>

   <!-- Definition of metadata element -->
   <xs:element name="metadata">
      <xs:annotation>
            <xs:documentation>metadata is used to store key-value pairs that
describe service results and data series</xs:documentation>
      </xs:annotation>
      <xs:complexType>
            <xs:simpleContent>
                  <xs:extension base="xs:string">
                        <xs:attribute name="key" type="xs:string"
use="required"/>
                  </xs:extension>
            </xs:simpleContent>
      </xs:complexType>
   </xs:element>
```

```
    <!-- Definition of dataValue element -->
    <xs:element name="dataValue">
        <xs:annotation>
              <xs:documentation>A set of data value elements is used to store
data values with their corresponding timestamps</xs:documentation>
        </xs:annotation>
        <xs:complexType>
              <xs:simpleContent>
                    <xs:extension base="xs:string">
                          <xs:attribute name="timestamp" type="xs:long"
use="required"/>
                    </xs:extension>
              </xs:simpleContent>
        </xs:complexType>
    </xs:element>

    </xs:schema>
```

APPENDIX C: WSDL OF CORRELATION WEB SERVICE

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="http://analyticservices.tut.fi/correlation"
xmlns:reqns="http://analyticservices.tut.fi/analyticService/request"
xmlns:respns="http://analyticservices.tut.fi/analyticService/response"
xmlns:sch="http://analyticservices.tut.fi/correlation"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://analyticservices.tut.fi/correlation"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"><wsdl:documentation>
</wsdl:documentation>

<wsdl:types>
<schema targetNamespace="http://analyticservices.tut.fi/correlation"
xmlns:reqns="http://analyticservices.tut.fi/analyticService/request"
xmlns:respns="http://analyticservices.tut.fi/analyticService/response"
xmlns:sch="http://analyticservices.tut.fi/correlation"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://analyticservices.tut.fi/correlation"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" elementFormDefault="qualified"
xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:import schemaLoca-
tion="http://localhost:8080/CorrelationService/services/AnalyticServiceCorrela
tionSoap11?xsd=request.xsd"
namespace="http://analyticservices.tut.fi/analyticService/request">
</xs:import>
<xs:import schemaLoca-
tion="http://localhost:8080/CorrelationService/services/AnalyticServiceCorrela
tionSoap11?xsd=response.xsd"
namespace="http://analyticservices.tut.fi/analyticService/response">
</xs:import>
</schema>
</wsdl:types>

<wsdl:message name="analyticServiceResponse">
<wsdl:part name="analyticServiceResponse" ele-
ment="respns:analyticServiceResponse">
</wsdl:part>
</wsdl:message>

<wsdl:message name="analyticServiceRequest"><wsdl:part
name="analyticServiceRequest" element="reqns:analyticServiceRequest">
</wsdl:part>
</wsdl:message>

<wsdl:portType name="AnalyticServiceCorrelation">
<wsdl:operation name="correlate">
<wsdl:input name="analyticServiceRequest" mes-
sage="tns:analyticServiceRequest">
 </wsdl:input>

<wsdl:output name="analyticServiceResponse" mes-
sage="tns:analyticServiceResponse">
</wsdl:output>
```

```
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="AnalyticServiceCorrelationSoap11"
type="tns:AnalyticServiceCorrelation">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http"
style="document"/>

<wsdl:operation name="correlate">
<soap:operation soapAction=""/>
<wsdl:input name="analyticServiceRequest">
<soap:body use="literal"/>
</wsdl:input>

<wsdl:output name="analyticServiceResponse">
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
</wsdl:binding>

<wsdl:service name="AnalyticServiceCorrelationService">
<wsdl:port name="AnalyticServiceCorrelationSoap11" bind-
ing="tns:AnalyticServiceCorrelationSoap11">
<soap:address loca-
tion="http://localhost:8080/CorrelationService/services/AnalyticServiceCorrela
tionSoap11"/>
</wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

APPENDIX D: REQUIREMENTS OF WEB SERVICES

Web Services	Requirements	Comments
Data Check	Any set KPI or Raw data.	Timestamp interval should be consistent throughout the KPI or Raw data.
Data correlation	Two sets of energy related KPI data series to be correlated	Number of data values in each set should be equal.
Data Normalization	Set of any KPI or raw data to be normalized	Optionally, normalization range can be provided.
Radar Chart	Energy related KPIs to be plotted and compared	Each KPI must have equal number of data values to be plotted.
Sankey Diagram	No KPI data. Only the energy input and losses data for each asset/process	KPI data is not required. Input and outputs are expressed in energy/power values.
Pareto Chart	Any Energy related KPI and the consumer assets/processes	Assets/processes should be the consumer of the KPI to be plotted on Pareto Chart.
Control Chart	Any KPI data or energy consumption raw data of the process	Number of values in each measurement should be the same.