



TAMPERE UNIVERSITY OF TECHNOLOGY

XIAOGUANG SONG
MONITORING ENERGY CONSUMPTION OF ARTIFICIAL
LIGHTING BY WEB INTERFACE
Master of Science Thesis

Supervisor: Docent Juha Miettinen
Dr.Tech. Lihong Yuan
The subject has been proved in
Faculty Council of the Faculty of
Automation, Mechanical and
Materials Engineering on 5th
September 2012.

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY

Master's Degree Programme in Machine Automation

Xiaoguang, Song: Monitoring Energy Consumption of Artificial Lighting by Web Interface

Master of Science Thesis, 94 pages, 8 Appendix pages

Major subject: Mechatronics

Examiner: Docent Juha Miettinen, Dr.Tech. Lihong Yuan

Keywords: web service, web application, complex event processing, ESPER

Study of energy efficiency has been very hot research topic in these days. Benefits from highly efficiency system are obvious. Greenhouse effect can be diminished, resources are protected and society development becomes sustainable. Development of information technology provides new solution to exiting problem from different fields. One of the application areas of information technology can be focused on monitoring energy performance of commercial building, which traditionally has been believed that this is one of main energy consumption sources.

Main object of this thesis work is to design a web application that is used to monitor energy usage of commercial building and provides reference standard on that particular room. The recommend energy usage is computed from standard EN-15193, which is drafted by EU and targeted to reduce unnecessary energy waste. Wireless sensor work has been utilized in this work for collecting environment data from field devices deployed across Tampere University of Technology. Adapter collects data from various different sources and forward to database located in company through Ethernet. Transmission control protocol/internet protocol connection is built up between host computer and company database. Raw data pass through filter energy predefined by user. Unrelated data is filtered out and relevant data are grouped and trigger action as long as data has satisfied specific conditions. At the same time, qualified data are stored within database which is managed by Hibernate framework. The Web application is developed from Spring MVC framework and is responsible for interaction between database and user interface. It also provides user interface to enter value of required parameters and displays final result. In this case, users do not need to install client application on their own computer, rather they can access to whole application by using web browser. Interaction between user and application is communicated through internet. Moreover, upgrade application process can be avoided on user side. As long as application upgrade is performed on server side, all client terminals are synchronized all together.

The result mainly shows functional web application from which information change between client and server is implemented. Energy usages from different control schemas are displayed on the Google chart. The energy analysis can be evaluated on hourly basis and daily basis. Finally, the optimal solution to particular room is derived from comparison.

摘要

坦佩雷理工大学

工厂自动化研究生学位，

宋晓光：基于网络应用的人工照明能量消耗的监控

研究生毕业论文,94 页，附录 8 页

主修：机电一体化

评审：副教授 **Miettinen Juha**，讲师 袁丽红

关键字：网络服务，网络应用，复杂事件处理

近些年，能量效率的研究已经成为热门话题。高效率系统的好处是显而易见的：包括温室气体排放的减少，自然能源的保护以及社会的可持续性发展。信息技术的发展提供了解决其他领域的新的方法。在过去，建筑的能量消耗被认为是重要的能源消耗源之一。如今商业化建筑的能量消耗的监控成为了其中一项重要的应用。

此篇论文主要研究的方向是设计一个网络应用去监测商业建筑物的能量消耗并提供能量消耗的参考值。能量消耗参考值是通过由 EN-15193 标准计算而得出的。这项标准是由欧盟制定的用于减少不必要的能量消耗。此篇论文包括了用于收集环境信息的无线网络的应用。此无线网络位于坦佩雷理工大学。适配器用于收集不同数据源的数据并通过以太网传输给位于公司的数据库。主机和公司的数据库通过 TCP/IP 连接。原始的数据通过用户设定的过滤器。不相关的数据将被过滤掉，相关的数据将被分组。当数据满足预设的条件，相关的数据处理机制将被激活。同时有用的数据会存入由 **Hibernate** 框架管理的数据库中。网络应用是根据 **Spring MVC** 框架搭建的，负责用户界面和数据库的交互。其中还包括提供给用户输入数据的交互界面。由于整个应用是基于网络的，所以用户不需要安装客户端在自己的电脑上。只要用户可以连接到互联网，用户就可以使用整个应用程序。此外用户不需要对应用程序在客户端进行升级。因为服务端的更新将会自动的与客户端同步。此篇论文的主要成果是设计了网络应用。客户端和服务器端可以经行有效的数据通讯。不同的控制机制的不同能量消耗可以通过 **Google Chart** 显示在客户端。能量分析可以既基于小时又可以基于天。最终，通过不同控制机制的能量消耗的比较，最优化的控制方案可以获得。

PREFACE

This thesis work is completed under the department of Mechanics and Design. To accomplish thesis work, I have met many problems and receive many help from my beloved friend. I want to express my gratitude to my supervisor, Docent Juha Miettinen, who offers me opportunity and support to finish my master thesis, Dr.Tech. Lihong Yuan, who assists me to complete my master thesis as adviser, and Zhang Bin who offers me the needed help.

My parents, Song Xinyi and Gaoyuan are always there and support me no matter what happens to me. Hence I want to dedicate my thesis work to them.

Living in Tampere is very exciting journey in my life. I want to express my grateful feeling to all my friends who have supported me.

Tampere, December 18th, 2012

Xiaoguang Song

TABLE OF CONTENTS

1.	Introduction.....	1
1.1.	Background	2
1.2.	Objectives.....	3
1.3.	Research methodology	3
1.4.	Thesis outline	4
2.	Literature and Technology Review	5
2.1	Concept of energy consumption standard	5
2.2	Methodology of energy consumption standard	5
2.3	Characteristic indicator of energy consumption standard	8
2.3.1	Daylight dependent artificial lighting control factor	8
2.3.2	Determination of daylight supply factor.....	9
2.3.3	Determination of the daylight dependency factor.....	12
2.3.4	Determine of occupancy dependency factor.....	15
2.3.5	Determination of constant illuminance factor.....	18
2.4	Web Service.....	22
2.4.1	Three tier architecture	23
2.4.2	Introduction to representational state transfer.....	24
2.4.3	Introduction to simple object access protocol.....	24
2.4.4	Comparison between REST and SOAP	24
2.5	Related work.....	27
3.	Research method.....	28
3.1	Occupancy sensor.....	28
3.2	Illuminance sensor.....	29
3.3	Wireless sensor network.....	30
3.4	Event Processing	32
3.4.1	Event processing engine	32
3.4.2	Introduction to implementation of event processing engine.....	33
3.5	Data base connection.....	34
3.5.1	Java database connectivity	35
3.5.2	Hibernate framework	36
3.6	Web application structure analysis	37
3.6.1	Model-view-control structure	37
3.6.2	Struts web application.....	39
3.6.3	Struts 2 web application.....	40
3.6.4	Spring model-view-control	42
3.6.5	Web application structure comparison.....	43
3.6.6	Integration of Spring framework with Hibernate framework	44
3.7	Spring Web Service	45
3.7.1	Web service definition language	45

3.7.2	Java architecture for XML binding	46
3.7.3	Inversion of control.....	47
3.7.4	Spring web service.....	48
3.8	Presentation technology	49
3.8.1	Java server pages.....	49
3.8.2	jQuery	49
3.8.3	Tiles.....	50
3.8.4	Google chart.....	51
3.8.5	Google Map	52
4.	Result	53
4.1	Introduction to test bed.....	53
4.2	Event processing of wireless sensor network.....	59
4.3	Implementation of Web application	61
5.	Conclusions.....	67
5.1	Conclusion on the thesis work.....	67
5.2	Future work	68
6.	References.....	70
A.	APPENDIX A – Script for activity sensors	74
B.	APPENDIX B – Script for illuminance sensors.....	75
C.	APPENDIX C – Script for instantaneous value of activity sensor	76
D.	APPENDIX D – Script for instantaneous value of illuminance sensor	77
E.	APPENDIX E – Script for configuration of delay time.....	78
F.	APPENDIX F – Script for energy consumption by manual control.....	79
G.	APPENDIX G – Script for energy consumption by automatic control	80
H.	APPENDIX H – Script for energy consumption by manual-automatic control	81

LIST OF FIGURES

Figure 1. Constitution of greenhouse gas emission.	1
Figure 2. Options to define energy usage.	6
Figure 3. Determination of daylight dependency factor.	13
Figure 4. Determine of the daylight dependency factor.	15
Figure 5. Determination of constant illuminance factor.	19
Figure 6. Structure of three tier architecture.	23
Figure 7. Communication mode of REST.	25
Figure 8. Communication mode of SOAP	25
Figure 9. Typical coverage of wall-mounted passive infrared sensor	29
Figure 10. TUT wireless sensor network infrastructure	30
Figure 11. client server communication sequence	31
Figure 12. Principle of event processing engines	33
Figure 13. Description of JDBC	35
Figure 14. Hibernate work principle.	37
Figure 15. Model-View-Controller model	38
Figure 16. Working procedure of Struts	40
Figure 17. Architecture of Struts2.	41
Figure 18. Work principle of Spring MVC	42
Figure 19. Comparison between WSDL 1.1 and WSDL 2.0.	46
Figure 20. Basic principle of JAXB	47
Figure 21. Principle of IOC.	47
Figure 22. Message flow in Spring Web Service	48
Figure 23 Principle of Ajax	50
Figure 24. Principle of Tiles	51
Figure 25. Principle of Google Chart	52
Figure 26. Layout of test bed	54
Figure 27. Layout of computer room in TUT.	56
Figure 28. Integrate wireless sensor.	57
Figure 29. Local map and geography coordinates.	62
Figure 30. Log in interface.	62
Figure 31. User interface for parameters input.	64
Figure 32. Historic energy consumption of different control schemas.	65
Figure 33. Energy consumption of different control schemas by hour.	66

LIST OF TABLES

Table 1. Default annual operating hours for building type.....	7
Table 2. Determination of artificial lighting control factor.	8
Table 3. Determination of direct hemispherical transmission of fenestration.	11
Table 4. Daylight penetration as function of the daylight factor	12
Table 5. Coefficients for determining daylight supply factor.....	12
Table 6. Monthly redistribution factor as function of daylight penetration	14
Table 7. Determination of FOC.....	16
Table 8. Determination of FA.....	17
Table 9. Value of LLMF for light source after hours of use	20
Table 10. Value of lamp survival factor for commonly light source	20
Table 11. Luminaire maintenance factor in clean environments	21
Table 12. Luminaire maintenance factor in normal environments	21
Table 13. Room surface maintenance factor in clean environment	22
Table 14. Room surface maintenance factor in normal environment	22
Table 15. Characteristics of simple object access protocol.	26
Table 16. Characteristics of representational state transfer.	26
Table 17. Functionality of illuminance sensor.....	29
Table 18. Comparison of EPL statement and match-recognize statements.....	34
Table 19. Comparison between Struts1 and Struts2 framework.....	43
Table 20. Comparison between Struts 2 and Spring framework.	44
Table 21. Sensor network distribution.....	55
Table 22. Dimension information and basic information about test bed room. ...	57
Table 23. Characteristic parameters of test bed room.....	58
Table 24. Important value of numerical indicators.....	59

LIST OF ABBREVIATIONS

AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
CEP	Complex Event Processing
DAO	Data Access Object
ESP	Event Stream Processing
SOAP	Simple Object Access Protocol
HTTP	Hypertext Transfer Protocol
HTML	Hyper Text Markup Language
HVAC	Heating, Ventilation, and air conditioning
IOC	Inversion of Control
IP	Internet protocol
JAXB	Java Architecture for XML Binding
JNDI	Java Naming and Directory Interface
JSON	JavaScript Object Notation
JSP	Java Server Page
LENI	Lighting Energy Numerical Indicator
LLMF	Lamp Lumen Maintenance Factor
LMF	Luminaries Maintenance Factor
LSF	Lamp Survival Factor
MAC	Media Access Control
MVC	Model, view control
PIR	Passive Infrared Sensor

POJO	Plain Old Java Object
REST	Representational State Transfer
RMI	Remote Method Invocation
RSMF	Room Surface Maintenance Factor
SIDF	Sensor Information Data Format
SQL	Structured Query Language
TUT	Tampere University of Technology
TCP	Transmission control protocol
WSDL	Web Service Definition Language
WSN	Wireless Sensor Network
WS	Web Service
XML	Extensible Markup Language

NOMENCLATURE

A	Total floor area of building
A_C	Area of facade opening [m ²]
A_D	Total area benefited from natural lighting [m ²]
a	Coefficient of determining daylight supply factor
$a_{D,ma}$	Maximum depth of daylight zone[m]
a_D	Depth of daylight zone[m]
b	Coefficient of determining daylight supply factor
b_D	Width of daylight zone[m]
$C_{D,S,n}$	Monthly distribution factor
D	Daylight factor for the zone [%]
F_A	Absence factor
F_c	Constant illuminance factor
F_D	Daylight dependency factor
$F_{D,S,n}$	Daylight supply factor
$F_{D,C,n}$	Daylight control factor
F_{OC}	Occupancy dependent light control system factor
F_{OC}	Occupancy dependent light control system factor
F_o	Occupancy dependency factor
h_{At}	Height from floor level to the top of atrium or courtyard[m]
h_{Li}	Height of lintel above floor[m]
H_m	Mounting height
h_{Ta}	Height of task area above floor[m]

I_O	Correction factor obstruction
$I_{O,OB}$	Correction factor for linear obstruction
$I_{O,OV}$	Correction factor for overhang
$I_{O,VF}$	Correction factor for vertical fins
$I_{O,CA}$	Correction factor for courtyard and atria
$I_{O,GDF}$	Correction factor for glazed double facade
$k_{AT,1}$	Factor regarding to frames of atrium roof
$k_{AT,2}$	Factor regarding to dirt on atrium roof
$k_{AT,3}$	Factor regarding to not normal light incidence
$k_{GDF,1}$	Factor accounting for frames of glazed double façade
$k_{GDF,2}$	Factor accounting for dirt of glazed double façade
$k_{GDF,3}$	Factor accounting for dirt of glazed double façade
k_1	Factor regarding to frame of fenestration system
k_2	Factor regarding to dirt on glazing
k_3	Factor regarding to not normal light incidence on façade
l_{At}	Length of atrium or courtyard[m]
L	Length of room
MF	Maintenance factor
P_{em}	Total charging power of emergency lighting in the room
P_n	Power of all luminaries in the room
P_{pc}	Total parasitic power of control in the room
RI	Room index
t_{At}	Transmission factor of atrium glazing for normal incidence

t_D	Daylight time usage
t_e	Emergency lighting charge time
t_N	Non daylight time usage
t_y	Standard year time
W	Total annual energy for lighting
$W_{i,d}$	Well-depth index
$W_{L,t}$	Energy consumption in period t
$W_{P,t}$	Parasitic energy consumption in period t
w	Width of room
w_{At}	Width of atrium or court yard[m]
$\gamma_{O,OB}$	Obstruction angle[°]
$\gamma_{O,OV}$	Obstruction angle [°]
$\gamma_{O,VF}$	Vertical fin angle [°]
γ_{Site}	Latitude angle of building location [°]
τ_{GDF}	Transmission factor of glazed double façade
τ_{D65}	Direct hemispherical transmission of fenestration

1. INTRODUCTION

Recently, global warming is drawn huge attention worldwide. It shows from Figure 1 that about almost 90 percent heat-trapping gas is emitted due to human activity [1]. Lighting and HVAC systems have been attractive targets for energy management in the building sectors, since together they are responsible for about 61% of a building's energy budget (lighting accounting for about 23%, with heating, ventilation and cooling accounting for about 38%) [2]. Gradually, lighting becomes a fast growing source of energy consumption and greenhouse gas emissions. Efficient use of lighting can decrease the lighting power waste, reduce the emission of greenhouse gases and other pollutants. The good news is that lighting systems provide huge opportunity for a cost effective energy saving with little or almost no inconvenience. In many situations, maintenance cost can be reduced meanwhile energy efficiency can be improved at the same time. There are three new challenges which drive the lighting industry to develop with fast pace; those are concerns for energy conservation, revolution of lamp technologies and higher standard on the quality of lighting source. Among those three drive forces, the requirement for high efficiency usage of lighting is the main motivation for developing new standard and system.

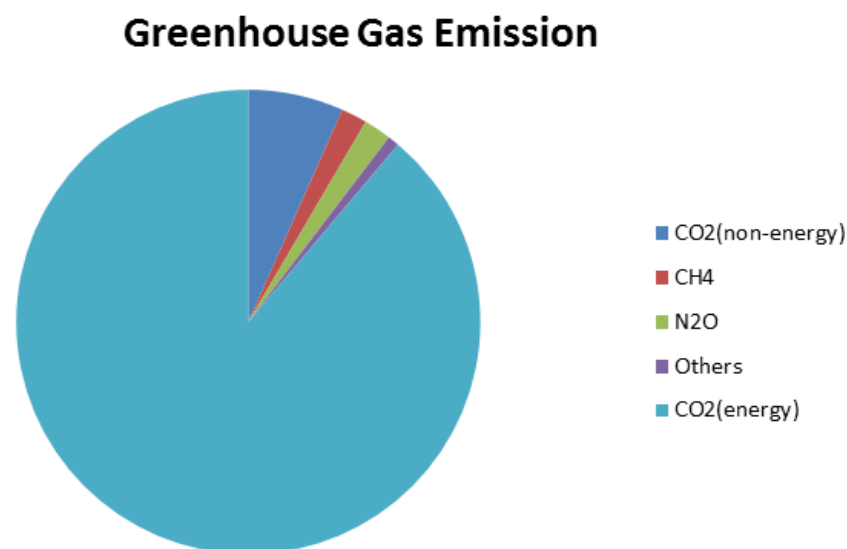


Figure 1. Constitution of greenhouse gas emission.

The classic approach to satisfy the requirements of standard is to provide appropriate amount of lighting at the right place and right moment. At the same time, harvesting of daylight can also improve it. So the main responsibility of good lighting system can turn off the light when users leave their activity area and switch on when user enter into zone at the right moment. Proper zone division plays important role in performance of lighting system. The compromise between area and resolution has to be reached. As to smaller zone area, it has greater control resolution, bigger energy saving effect and keeps balance between demand and supply more precisely. But complexity and cost of control system increases, since more control devices are required installed at the field. Appropriate control schema can attain the goal of providing light when user need. Two main categories of control schema are schedule based and sensor based control. As to the schedule driven control schema, it suits to the time pattern of occupation of spaces which is fixed or easily predicted, so that control system can be pre-programmed. On the other hand, occupancy detection in many buildings is used to reduce energy consumption [3]. This is typically implemented by occupancy sensors, which are connected to each other to form network.

1.1. Background

The building sector in the EU area is using 40% of the total EU energy consumption and is responsible for 36% of the CO₂ emissions[4]. Commercial buildings, primarily office building, are viewed as the highest energy consumption sector. The total annual energy usage in the office buildings varies from 100 to 1000 kWh/m² yr, depending on geographic location, use and type of office equipment, operational schedules, type of envelop, use of HVAC systems, type of lighting etc [5]. In North Europe, office energy intensity lies in the range 269-350kWh/m² yr, and for offices in all over Europe, it is about 306kWh/m² yr, with mean electric index 150 kWh/m² yr and mean fuel index 158 kWh/m² yr [6]. European Union has actively promoted the campaigns for energy efficiency and renewable energy [7].

Promotion of appropriate standard related to lighting usage in the building becomes paramount important. This standard is derived from good lighting practice and give guidance on regulation of newly built or retrofit commercial building. Procedure and convention for estimation of energy requirement in building is drafted by European countries due to those motivations. The numerical indicators which represent energy efficiency are defined. The standard for different types of activities and space to guarantee energy efficiency is prescribed and preference of certain technology or

control system is recommended. In the standard, it also provides reference for evaluation of performance of commercial building.

The standard EN 15193 has specifies calculation method for evaluation of energy consumption of building lighting [8]. Correct lighting standard for building energy consumption is of most importance for sake of energy saving. Procedure and convention assures that lighting control scheme complies with good practice. This standard provides procedure both for estimation of energy requirements of building lighting and methodology for numeric indicator for energy performance of the commercial building. If the resolution of the occupancy sensor is qualified, it can offer better performance.

Traditionally, task of monitoring of performance of energy consumption in the building is fulfilled by building automation system (BAS). However, two challenges occur. Integration of subsystem from different vendors into existing system is still quite difficult due to the proprietary solution. Sometimes integrated system not only does not benefit from upgrade, but some of their functions are also suffered. The conflict between BAS and enterprise application is proved to be the other challenge. Those two problems can be overcome by adapting web application, which offers the flexibility.

1.2. Objectives

Two objects of this thesis are 1) to estimate energy requirement of lighting in commercial buildings; and 2) to monitor energy consumption by building the wireless network and displaying result on webpage by web application. The effects of different control schema are compared and optimal control schema for that specific environment is obtained. Due to lack of actuator, control action is simulated by algorithm.

1.3. Research methodology

This thesis work consists of two sections. In the first section, standard of evaluation of energy consumption of lighting in commercial building is described. The main formulas are explained and main characteristic factors are studies. The estimation value of individual room is obtained. In the second part of the thesis, sensor network and web application is implemented. All sensors are deployed and the wireless sensor network is connected. The data fetched from sensor network is filtered first according to predefine rule and then stored in data base for web application. After comparison of advantages and disadvantages of different solution, appropriate design is selected and implemented. The final result will be displayed on browser at the client side.

1.4. Thesis outline

This thesis work is organized as follows: Chapter 2 will briefly describe the literature review on the standard of BS EN 15193:2007 including calculation formula, and important numerical indicators. Chapter 3 introduces occupancy sensor, wireless sensor network, and flow pattern of acquisition for relevant date. Design and implementation of web application for monitoring energy consumption in the commercial buildings are also included in this chapter. The description of data analysis of use case and final result is presented in the Chapter 4. The final conclusion and future work is depicted in the Chapter 5.

2. LITERATURE AND TECHNOLOGY REVIEW

Before going deep into detail of research method, literature and technology review is helpful to determine the direction of research. In this section, energy consumption standard, BS EN 15193, is introduced. Methodology of BS EN 15193 is reviewed. After that, web application structure is presented. Related work completed by other research is discussed.

2.1 Concept of energy consumption standard

BS EN 15193 which is British standard on energy performance of building was published in 2007 as part of EU action to meet the Kyoto Agreement regarding to reduction of greenhouse gas by increasing energy efficiency. This standard focuses on actual energy consumption, instead of just installed power. This standard estimation also takes quality of light source into consideration, when it calculates energy requirement. The lighting energy numeric indicator can be derived from standard, which is used to compare energy usage of similar type of building with different size and configuration.

There are some assumptions about energy calculation as following:

- 1) Parasitic powers not contained in luminaire are ignored.
- 2) Lighting supplies for facade illumination, such as car parking lighting, security lighting are not taken into account.

2.2 Methodology of energy consumption standard

Estimation of lighting energy requirement consists of two main categories, calculated estimation and metered evaluation. The latter one includes measurement of lighting circuits in electrical distribution and controller of lighting management system. As to former one, energy requirement is divided into two types, luminaire power and parasitic power. Luminaire power is used to provide lighting for functional illumination which conforms to EN 12193 [9] for sport facilities and EN 12464-1 [10] for indoor work places. Parasitic power is energy supply for lighting control system and charging batteries for emergency lighting which conforms to EN 1838 [11]. Design of luminaires and its component is in accordance with EN 60598 [12], EN 60570 [13] and EN 61347 [14]. As to the calculation method, it also provides two different set of calculation formula, quick method and comprehensive method. The energy requirement estimated by the quick method usually has higher value than by accurate comprehensive method.

In this thesis work, the comprehensive calculation method is explained. Procedure of comprehensive calculation mythology is described in Figure 2 [15].

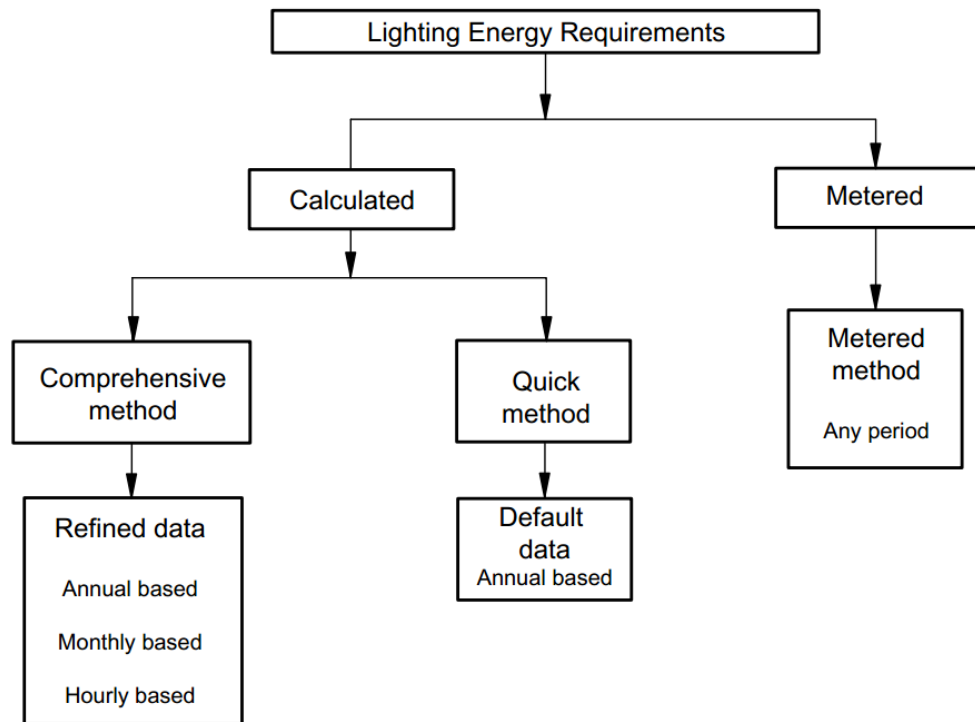


Figure 2. Options to define energy usage.

The total estimated energy for the room within given time can be calculated by the equation [15]:

$$W_t = W_{L,t} + W_{P,t} [kWh] \quad (2-1)$$

Where $W_{L,t}$ is energy consumption in period t, $W_{P,t}$ is parasitic energy consumption in period t


Energy consumption for meeting the illumination requirement of building is estimated by [15]:

$$W_{L,t} = \sum \{ (P_n \cdot F_c) \cdot [(t_D \cdot F_o \cdot F_D) + (t_N \cdot F_o)] \} / 1000 [kWh] \quad (2-2)$$

Where P_n is power of all luminaries in the room. F_c is constant illuminance factor, t_D is daylight time usage, F_o is occupancy dependency factor, F_D is daylight dependency factor, t_N is non daylight time usage

From BS EN 15193:2007 standard, it provides default value of daylight time usage, non-daylight time usage and annual operating hour. Those values are stored and looked up from database. This can be seen in Table 1, which is snap shot of database implementation. This information is stored into database for the web application.

Table 1. Default annual operating hours for building type.

 id *	building_type	usagetime	factor
▶ 1	offices	daytime	2250
2	offices	nighttime	250
3	offices	operatehour	2500
4	education	daytime	1800
5	education	nighttime	200
6	education	operatehour	2000
7	hospital	daytime	3000
8	hospital	nighttime	2000
9	hospital	operatettime	5000
10	hotel	daytime	3000
11	hotel	nighttime	2000
12	hotel	operatettime	5000
13	restaurant	daytime	1250
14	restaurant	nighttime	1250
15	restaurant	operatettime	2500
16	sport	daytime	2000
17	sport	nighttime	2000
18	sport	operatettime	4000
19	retail	daytime	3000
20	retail	nighttime	2000
21	retail	operatettime	5000
22	factory	daytime	2500
23	factory	nighttime	1500
24	factory	operatettime	4000

Estimation of parasitic energy including charging energy for standby energy and emergency lighting can be determined by [15]:

$$W_{P,t} = \sum \left\{ \left\{ P_{pc} \cdot [t_y - (t_D + t_N)] \right\} + (P_{em} \cdot t_e) \right\} / 1000 \text{ [kWh]} \quad (2-3)$$

Where P_{pc} is total parasitic power of control in the room, t_y is standard year time, P_{em} is total charging power of emergency lighting in the room, t_e is emergency lighting charge time.

Lighting Energy Numeric Indicator (LENI) for building is defined by [15]:

$$LENI = W/A \text{ [kWh}/(m^2 \cdot \text{year})] \quad (2-4)$$

Where W is total annual energy for lighting, A is total floor area of building.


2.3 Characteristic indicator of energy consumption standard

The core important component of BS EN 15193 is to investigate characteristic indicator of specific building, which value reflect configuration of that room. Among them, there are three most important factors, constant illuminance factor, occupancy dependency factor, and daylight dependency factor. Procedure of estimation of those three factors is illustrated in this section.

2.3.1 Daylight dependent artificial lighting control factor

Daylight dependent artificial lighting control factor is used to describe the efficiency towards different types of control strategy. From Table 2, it shows correction factor of daylight supply. This figure is obtained from database table. The position of installed devices has big impact on accuracy of this correction factor. This factor table will be stored into database for web application to fetch relevant value. Once user log in the system, business layer of web application will automatically search for this information and calculate daylight dependency factor.

Table 2. Determination of artificial lighting control factor.

 id *	control	penetration	factor
▶ 1	manual	weak	0.2
2	manual	medium	0.3
3	manual	strong	0.4
4	auto	weak	0.75
5	auto	medium	0.77
6	auto	strong	0.85

2.3.2 Determination of daylight supply factor

Space area of commercial building is divided into daylight zone and non-daylight zone. If area of daylight zone is contributed by several facades, total area of daylight zone is the sum of individual area. The maximum depth of zone which receives light from facades is expressed as [15]:

$$a_{D,max} = 2.5 \cdot (h_{Li} - h_{Ta}) \quad (2-5)$$

Where $a_{D,max}$ is maximum depth of daylight zone[m], h_{Li} is height of lintel above floor[m], h_{Ta} is height of task area above floor[m]

The depth of daylight zone is measured from interior surface of exterior wall, perpendicular towards facade. Area of daylight space is as following [15]:

$$A_{Dj} = a_D \cdot b_D \quad (2-6)$$

Where a_D is depth of daylight zone[m], b_D is width of daylight zone[m].

Width of daylight zone can be defined by interior width of facade of building zone. If only some section of facade contains window, width of daylight zone equals addition of width of facade containing windows and half of depth of daylight zone.

Daylight supply factor are attributed by two factors, vertical facade and roof light. In this thesis work, offices and classroom of university are the research targets, in which factor of roof light is neglected. Space of commercial building benefiting from daylight is dependent on geometric conditions. Those geometric conditions are transparency index, depth index, and obstruction index, which are explained below.

Firstly, transparency index indicates benefit obtained from daylight and it is written as [15]:

$$I_t = A_C / A_D \quad (2-7)$$

Where A_C is area of facade opening [m²], A_D is total area on horizontal surface benefited from natural lighting [m²]

Secondly, depth index of space is expressed as [15]:

$$I_{De} = a_D / (h_{Li} - h_{Ta}) \quad (2-8)$$

Thirdly, obstruction index describes effect reducing light incident onto facade. This factor accounts for the effect of obstruction, such as from natural obstacle, courtyard or atrium design, overhangs of facade and glazed double facade. It can be defined as [15]:

$$I_O = I_{O,OB} \cdot I_{O,OV} \cdot I_{O,VF} \cdot I_{O,CA} \cdot I_{O,GDF} \quad (2-9)$$

Where I_O is correction factor obstruction, $I_{O,OB}$ is correction factor for linear obstruction,

$I_{O,OV}$ is correction factor for overhang, $I_{O,VF}$ is correction factor for vertical fins, $I_{O,CA}$ is correction factor for courtyard and atria, $I_{O,GDF}$ is correction factor for glazed double facade

In the above formula, correction factor for linear obstruction is defined as [15]:

$$\begin{aligned} I_{O,OB} &= \cos(1.5 \cdot \gamma_{O,OB}) & \gamma_{O,OB} < 60^\circ \\ I_{O,OB} &= 0 & \gamma_{O,OB} \gg 60^\circ \end{aligned} \quad (2-10)$$

Where $\gamma_{O,OB}$ is obstruction angle($^\circ$)

Correction factor for overhang is defined as [15]:

$$\begin{aligned} I_{O,OV} &= \cos(1.33 \cdot \gamma_{O,OV}) & \gamma_{O,OV} < 67.5^\circ \\ I_{O,OV} &= 0 & \gamma_{O,OV} \gg 67.5^\circ \end{aligned} \quad (2-11)$$

Where $\gamma_{O,OV}$ is obstruction angle [$^\circ$]

Correction factor for vertical fins is obtained by [15]:

$$I_{O,OF} = 1 - \gamma_{O,VF}/300^\circ \quad (2-12)$$

Where $\gamma_{O,VF}$ is vertical fin angle [$^\circ$]

Well-depth index is listed as [15]:

$$w_{i_d} = h_{At} \cdot (l_{At} + w_{At}) / (2 \cdot I_{At} \cdot w_{At}) \quad (2-13)$$

Where W_{i_d} is well-depth index, h_{At} is height from floor level to the top of atrium or courtyard[m], l_{At} is length of atrium or courtyard[m], w_{At} is width of atrium or courtyard[m]

Correction factor for courtyard and atria is obtained by [15]:

For courtyard

$$I_{O,CA} = 1 - 0.85 \cdot w_{i_d} \quad (2-14)$$

For atria, it is defined as [15]:

$$\begin{aligned} I_{O,CA} &= \tau_{At} \cdot k_{AT,1} \cdot k_{AT,2} \cdot k_{AT,3} \cdot (1 - 0.85 \cdot w_{i_d}) \\ I_{O,CA} &= 0 \text{ for } w_{i_d} > 1.18 \end{aligned} \quad (2-15)$$

Where $k_{AT,1}$ is factor regarding to frames of atrium roof, $k_{AT,2}$ is factor regarding to dirt on atrium roof, $k_{AT,3}$ is factor regarding to not normal light incidence on atrium roof, t_{At} is transmission factor of atrium glazing for normal incidence

Correction factor for glazed double facade is defined by [15]:

$$I_{O,GDF} = \tau_{GDF} \cdot k_{GDF,1} \cdot k_{GDF,2} \cdot k_{GDF,3} \quad (2-16)$$

Where τ_{GDF} is transmission factor of glazed double façade, $k_{GDF,1}$ is factor accounting for frames of glazed double façade, $k_{GDF,2}$ is factor accounting for dirt of glazed double façade, $k_{GDF,3}$ is factor accounting for dirt of glazed double façade.

Daylight factor for facade opening factor is defined by [15]:

$$D_C = (4.13 \cdot 20.0 \cdot I_T - 1.36 \cdot I_{De}) \cdot I_O(\%) \quad (2-17)$$

Daylight factor for the zone which describes relationship between energy demand and fenestration property is expressed as [15]:

$$D = D_C \cdot \tau_{D65} \cdot k_1 \cdot k_2 \cdot k_3[\%] \quad (2-18)$$

Where D is daylight factor for the zone [%], τ_{D65} is direct hemispherical transmission of fenestration, k_1 is factor regarding to frame of fenestration system, k_2 is factor regarding to dirt on glazing, k_3 is factor regarding to not normal light incidence on façade

Direct hemispherical transmission of fenestration factor is defined by Table 3, which is stored in the database. Transmission of fenestration factor is highly dependent on the type of windows. Single glazing, double glazing, triple glazing has big difference in property of fenestration. Difference also exists among different type of solar glazing.

Table 3. Determination of direct hemispherical transmission of fenestration.

id *	type	fenestration
1	single_glazing	0.9
2	double_glazing	0.82
3	triple_glazing	0.75
4	low_e1_double	0.74
5	low_e2_double	0.78
6	low_e3_double	0.78
7	low_e1_triple	0.69
8	low_e2_triple	0.69
9	solar_glazing_1	0.59
10	solar_glazing_2	0.67
11	solar_glazing_3	0.4
12	double_glazing	0.82
13	triple_glazing	0.75
14	low_e1_double	0.74
15	low_e2_double	0.78
16	low_e3_double	0.78
17	low_e1_triple	0.69
18	low_e2_triple	0.69
19	solar_glazing_1	0.59
20	solar_glazing_2	0.67
21	solar_glazing_3	0.4

Daylight factors for facade opening and daylight factor for the zone indicate impact of fenestration system on daylight penetration and is expressed in Table 4 [15].

Table 4. Daylight penetration as function of the daylight factor.

Classification		Daylight penetration (access of the zone to daylight)
D_c	D	
$D_c \geq 6\%$	$D \geq 3\%$	Strong
$6\% > D_c \geq 4\%$	$3\% > D \geq 2\%$	Medium
$4\% > D_c \geq 2\%$	$2\% > D \geq 1\%$	Weak
$D_c < 2\%$	$D < 1\%$	None

Finally, the daylight supply factor can be expressed as [15]:

$$F_{D,S} = a + b \cdot \gamma_{Site} \quad (2-19)$$

Where a is coefficient of determining daylight supply factor, b is coefficient of determining daylight supply factor, γ_{Site} is latitude angle of building location [$^{\circ}$]

The coefficients of determining daylight supply factor are determined according to different maintained illuminance and daylight penetration, which is listed in Table 5. In this table, detail information about database implementation is displayed.

Table 5. Coefficients for determining daylight supply factor.

id *	illuminance	penetration	factora	factorb
1	300	weak	1.2425	0.0117
2	300	medium	1.3097	0.0106
3	300	strong	1.2904	0.0088
4	500	weak	0.9432	0.0094
5	500	medium	1.2425	0.0117
6	500	strong	1.322	0.011
7	750	weak	0.6692	0.0067
8	750	medium	1.0054	0.0098
9	750	strong	1.2812	0.0121

2.3.3 Determination of the daylight dependency factor

Daylight dependency factor is function of daylight supply factor and control factor of daylight dependent artificial lighting control factor. To calculate the value of daylight dependency factor, segmentation of building into area with and without daylight access is

defined. Characteristics of commercial building are studies, such as room dimension, facade geometry, exterior obstruction, and daylight penetration. Its process is shown in Figure 3 [16]. Daylight supply factor which is used to describe energy saving potential of building is determined by defining local climate condition and maintained illuminance requirement. Daylight control factor which describes the relationship between consumption of installed lighting power and daylight availability.

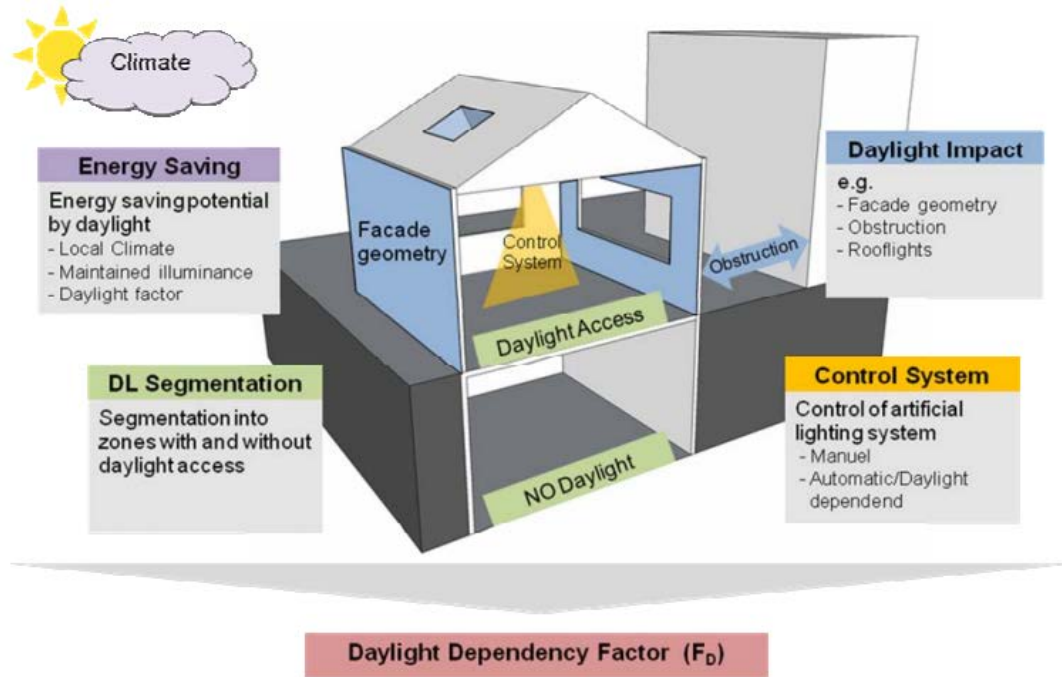


Figure 3. Determination of daylight dependency factor.

Daylight dependency factor can be calculated by following formula [15].

$$F_{D,n} = 1 - (F_{D,S,n} \cdot F_{D,C,n}) \quad (2-20)$$

Where $F_{D,S,n}$ is daylight supply factor, $F_{D,C,n}$ is daylight control factor.

As refer to the conversion daylight dependency factor into monthly based factor, it can be expressed by[15]:

$$F_{D,n} = 1 - (F_{D,S,n} \cdot F_{D,C,n} \cdot C_{D,S,n}) \quad (2-21)$$

Where $C_{D,S,n}$ is monthly distribution factor

Monthly distribution factor is chosen from Table 6 [15]. This factor is used to convert annual basis energy requirement into monthly basis.

Table 6. Monthly redistribution factor as function of daylight penetration.

Weather station / location		Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Watford, GB 52°N	weak	0,38	0,68	1,02	1,36	1,56	1,62	1,53	1,39	1,13	0,77	0,28	0,28
	medium	0,47	0,80	1,05	1,30	1,46	1,42	1,40	1,35	1,16	0,89	0,35	0,35
	strong	0,61	0,88	1,07	1,24	1,30	1,28	1,28	1,28	1,16	0,97	0,47	0,47
Frankfurt, D 50°N	weak	0,43	0,65	0,94	1,33	1,46	1,58	1,55	1,41	1,08	0,76	0,46	0,34
	medium	0,50	0,73	1,01	1,28	1,38	1,44	1,43	1,35	1,11	0,83	0,53	0,40
	strong	0,62	0,84	1,07	1,21	1,27	1,28	1,28	1,25	1,12	0,91	0,64	0,51
Athens, GR 38°N	weak	0,65	0,87	1,08	1,22	1,25	1,17	1,24	1,20	1,04	0,93	0,75	0,60
	medium	0,74	0,91	1,05	1,13	1,17	1,15	1,19	1,14	1,05	0,95	0,81	0,69
	strong	0,83	0,97	1,05	1,09	1,10	1,10	1,10	1,08	1,05	0,97	0,87	0,78
Bratislava, SK 48°N	weak	0,45	0,79	1,02	1,34	1,41	1,51	1,40	1,37	1,05	0,83	0,48	0,35
	medium	0,54	0,88	1,05	1,25	1,32	1,37	1,32	1,29	1,08	0,91	0,57	0,43
	strong	0,65	0,94	1,06	1,18	1,23	1,24	1,23	1,21	1,08	0,95	0,67	0,54
Lyon, F 46°N	weak	0,49	0,74	1,09	1,26	1,35	1,41	1,38	1,31	1,09	0,87	0,56	0,42
	medium	0,59	0,84	1,11	1,21	1,25	1,27	1,26	1,25	1,11	0,94	0,66	0,51
	strong	0,70	0,92	1,10	1,14	1,17	1,16	1,17	1,17	1,10	0,98	0,76	0,63
Gavle, S 61°N	weak	0,21	0,55	1,04	1,45	1,62	1,73	1,68	1,55	1,10	0,65	0,27	0,12
	medium	0,25	0,65	1,12	1,42	1,53	1,57	1,56	1,51	1,16	0,75	0,33	0,15
	strong	0,32	0,77	1,17	1,35	1,44	1,44	1,45	1,42	1,19	0,84	0,42	0,19

The whole procedure for determination of daylight dependency factor can be described as following flow chart in the Figure 4 [9]. At beginning, daylight penetration is studied to determine if it is weak, medium or strong. This is studied by analysing obstruction index, transparency index, depth index, daylight penetration. This result leads to value of daylight supply factor. Monthly supply factor can be derived from daylight supply factor by monthly method and operating hours. If there is no daylight penetration for this room, value of daylight dependency factor is assign to one so that daylight is not relate with illuminance condition within this room. This method is also possible to calculate annual daylight dependency factor.

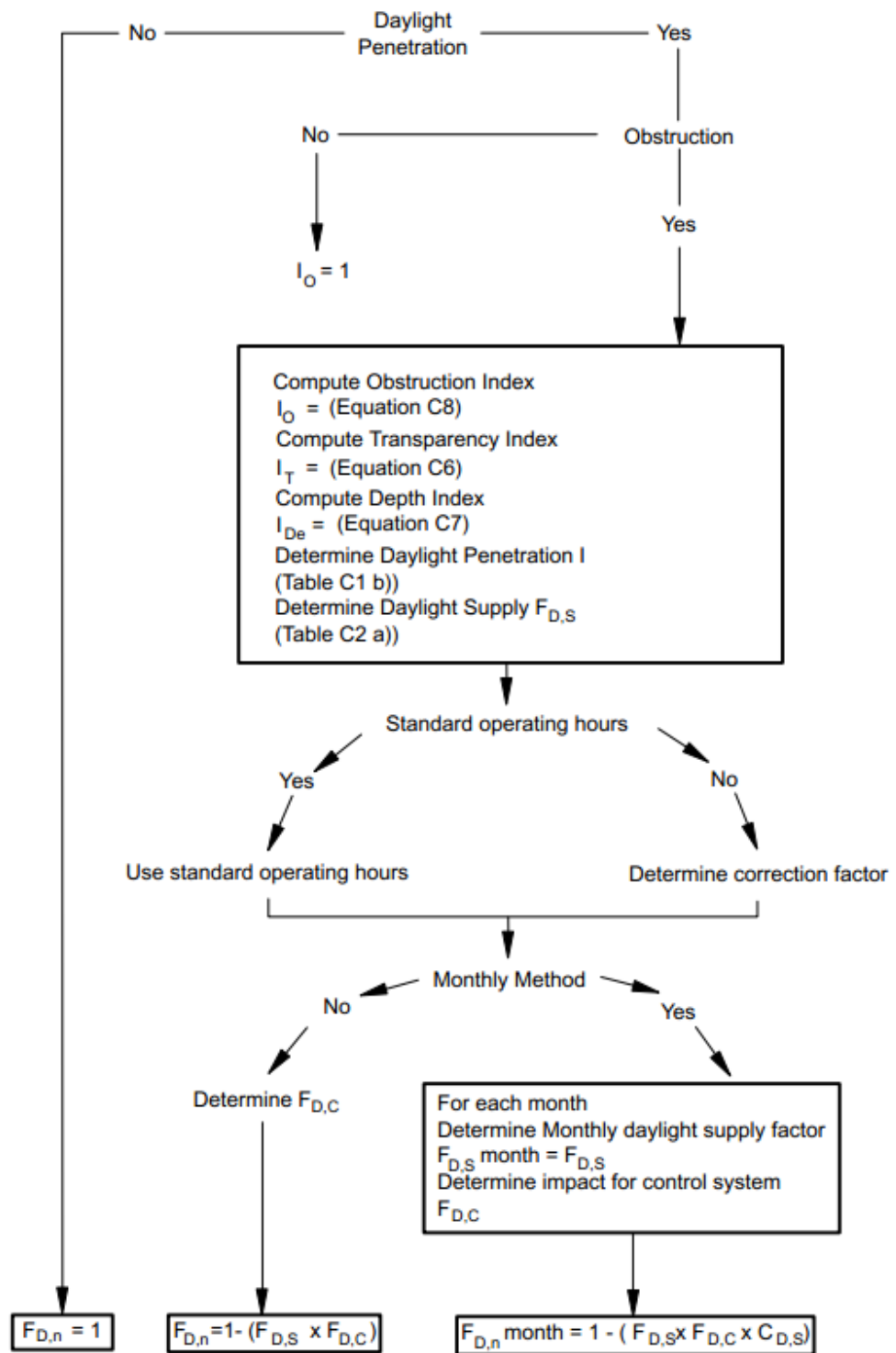


Figure 4. Determine of the daylight dependency factor.

2.3.4 Determine of occupancy dependency factor

Occupancy dependency factor is to estimate of energy efficiency of lighting control system, which depends on type of control system and presence and absence of the room.



The value of this factor is reverse proportional to the occupancy time. Occupancy factor can be considered as one, if lighting is certainly on. This means that whole area of building is controlled by one automation system or illuminated area controlled by one schema is larger than 30 m². In other case, occupancy dependency factor is varied according to absence factor. If absence factor is less than 0.2, occupancy dependency factor is close to one. If absence factor is between 0.2 and 0.9, slope of different curve is identical. If absence factor is great than 0.9, occupancy dependency factor is close to zero. Formulas for different cases are shown below [15].

$$\begin{aligned}
 \text{If } 0.0 \leq F_A < 0.2 \quad F_O &= 1 - [(1 - F_{OC}) \cdot F_A / 0.2] \\
 \text{If } 0.2 \leq F_A < 0.9 \quad F_O &= F_{OC} + 0.2 - F_A \\
 \text{If } 0.9 \leq F_A < 1.0 \quad F_O &= [7 - (10 \cdot F_{OC})] \cdot (F_A - 1)
 \end{aligned} \tag{2-22}$$

Where F_{OC} is occupancy dependent light control system factor, F_A is absence factor, which is proportional to the time that space is occupied

The occupancy dependent light control system factor is function of lighting control system. This factor value is defined within Table 7, which is part of tables of database.

Table 7. Determination of FOC.

	 id *	controlschema	factor
	1	manual_on_off	1
	2	auto_on_dim	0.95
	3	auto_on_off	0.9
	4	manual_on_dim	0.9
	5	manual_on_au...	0.8

Absence factor is percentage of room occupied. In the Table 8, different values of absence factors are presented according to building type and room. Those data are stored in the database, which can be accessed by connection to database of application. Buildings are grouped into four categories, office, educational building, hospital and factory. Within each type of building, different types of room are assigned different value of absence factor.

Table 8. Determination of F_A .

id	building_type	room_type	factor
1	Offices	cellular_office_1	2.00E-01
2	Offices	cellular_office_2-6	3.00E-01
3	Offices	open_plan_office_30	0.00E+00
4	Offices	open_plan_office_10	2.00E-01
5	Offices	corridor	4.00E-01
6	Offices	entrance_hall	0.00E+00
7	Offices	show_room	6.00E-01
8	Offices	bathroom	9.00E-01
9	Offices	rest_room	5.00E-01
10	Offices	storage_room	9.00E-01
11	Offices	technical_plant_room	9.80E-01
12	Offices	copying_room	5.00E-01
13	Offices	conference_room	5.00E-01
14	Offices	archives	9.80E-01
15	Educational	classroom	2.50E-01
16	Educational	group_room	3.00E-01
17	Educational	corridor	6.00E-01
18	Educational	junior_room	5.00E-01
19	Educational	lecture	4.00E-01
20	Eudcational	sport	3.00E-01
21	Educational	dining_hall	2.00E-01
22	Educational	staff_room	4.00E-01
23	Educational	copying_room	4.00E-01
24	Educational	kitchen	2.00E-01
25	Educational	library	4.00E-01
26	Hospital	Wards	0.00E+00
27	Hospital	examination	4.00E-01
28	Hospital	preoperation	4.00E-01
29	Hospital	recovery_room	0.00E+00
30	Hospital	operating_theatre	0.00E+00
31	Hospital	corridor	0.00E+00
32	Hospital	conduit	7.00E-01
33	Hospital	waitingarea	0.00E+00
34	Hospital	entrance_hall	0.00E+00
35	Hospital	day_room	2.00E-01
36	Hospital	laboratory	2.00E-01
37	factory	assembly	0.00E+00
38	factory	small_assembly	2.00E-01
39	factory	strorage	4.00E-01
40	factory	open_storage	2.00E-01
41	factory	paintting_room	2.00E-01

As to the system without occupancy sensor, lighting devices are controlled by manual

switch within the room. As to the automatic system, it is divided into four different control schema, Auto on/ Dimmed, Auto On/ Auto off, Manual On/ Dimmed, and Manual On/ Auto off. For the first control schema, control system switches light on as long as presence of user is detected. The power output decreases no later than 15minutes after last detection of presence in the room. As to the second one, lighting will switch on whenever presence is detected and totally switched off no later than 15minutes after last detection of presence in the room. As to the third one, lighting is switched on by manual switch and it will decrease power output to no more than 20 percent of operative state no later than 15 minutes after the last detection of presence in the room. As to the last one, System turned on light by manual switch and turned off no later than 15 minutes after last detection of presence in the room.

There are several other schemas that improve the performance of control system. One is to utilize the combination of manual on/off schema with automatic sweeping extinction action, which prevents lighting energy waste during night time. The other is to adapt automatic on and manual off schema. By this combination, the detection system can also be switched off with lighting source. In this way, parasitic power can be dramatically reduced. This method can also avoid unnecessary energy consumption due to eliminated out the possibility of misjudgement of entry action. The system will not be switched on lighting because detection of presence of neighbour area is avoided and short time entry is not misunderstood as room is occupied.

2.3.5 Determination of constant illuminance factor

As to dimmable lighting system, automatic control system can decrease initial output of lighting source and increase the output with the time. This type of system is so called controlled constant illuminance system. Once installed power is equal to the power demand, the lighting source is needed maintenance, such as replacing with new device, and cleaning luminaire or the room. The power of constant illuminance factor is the ratio of actual input power at given time to initial installed power. So the constant illuminance factor is ratio of average power at given time to initial installed power. This factor can be defined as [15]:

$$F_c = (1 + MF)/2 \quad (2-23)$$

Where MF is maintenance factor

Constant illuminance factor depends on two variables, power and maintenance factor. To maintain the constant illuminance level, power of lighting fixture will not delivery maximum power. The increasing output power of lighting fixture changes over time, since lighting power decay. In this case, maintain factor is keeping downward. This whole process is depicted in the Figure 5[17].

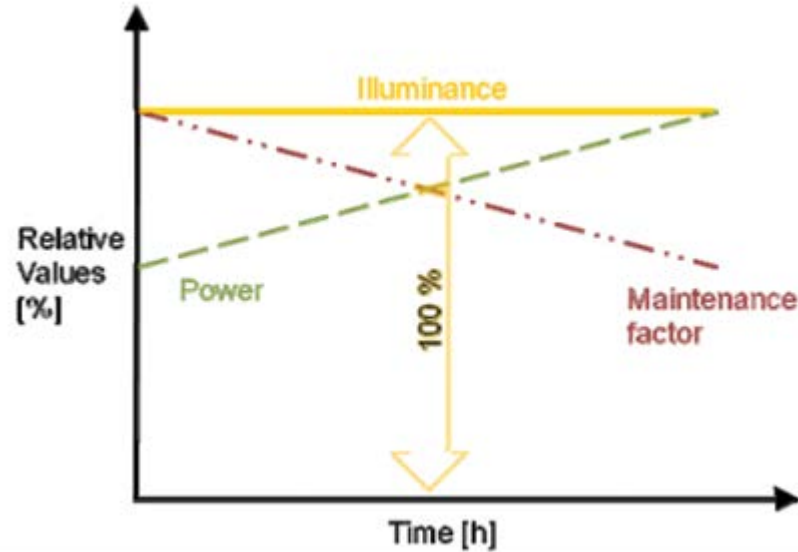


Figure 5. Determination of constant illuminance factor.

Maintenance requirement for lighting device should be considered at design stage already, since the initial installation. Illuminance devices usually start to decay which means output of devices decrease and energy consumption increases at the same time to maintain initial illuminance level. This deteriorative rate of equipment is expressed by maintenance factor, which is ratio of maintained illuminance and initial illuminance. As to guarantee the consistency of illuminance level, initial illuminance level provided by system should be $1/MF$ greater than initial condition. The maintenance factor can be expressed as [17]:

$$MF = LLMF \cdot LSF \cdot LMF \cdot RSMF \quad (2-24)$$

Where MF is maintenance factor, $LLMF$ is lamp lumen maintenance factor, LSF is lamp survival factor, LMF is luminaire maintenance factor, $RSMF$ is room surface maintenance factor.

Luminous flux from lighting device decreases with time of operation. The decline rate depends on type of light source. Lamp lumen maintenance factor is used to describe operation hour for specific device, which is proportional to initial output after specific operation time. This factor can be defined as reduction percentage per thousand hours of operation. This data is offered by manufactures based on British Standard test. In this test, ambient temperature, applied voltage which affects the performance of lighting supply is taken into consideration. Typical value of $LLMF$ is listed as Table 9 [17].

Table 9. Value of LLMF for light source after hours of use.

Light source	Hours of use (thousands)										
	0.1	0.5	1.0	1.5	2	4	6	8	10	12	14
Triphosphor/multiphosphor fluorescent	1	0.98	0.96	0.95	0.94	0.91	0.87	0.86	0.85	0.84	0.83
Halophosphor fluorescent	1	0.97	0.94	0.91	0.89	0.83	0.80	0.78	0.76	0.74	0.72
Mercury	1	0.99	0.97	0.95	0.93	0.87	0.80	0.76	0.72	0.68	0.64
High pressure sodium	1	1	0.98	0.97	0.96	0.93	0.91	0.89	0.88	0.87	0.86
Improved colour high pressure sodium	1	0.99	0.97	0.95	0.94	0.89	0.84	0.81	0.79	0.78	-
Low pressure sodium	1	1	0.99	0.98	0.98	0.98	0.97	0.97	0.96	0.96	0.96

Lamp survival factor is described as the percentage of lighting equipment of specific type that can sustain emitting light after certain duration of operation. This factor is provided from manufacturer and this factor based on supply voltage, control gear and switching cycle. If the actual operative environment is different from the suggested condition, the lamp survival factor needs to be reevaluated. The recommend value of lamp survival factor is listed in the Table 10 [17].

Table 10. Value of lamp survival factor for commonly light source.

Light source	Hours of use (thousands)										
	0.1	0.5	1.0	1.5	2	4	6	8	10	12	14
Triphosphor/multiphosphor fluorescent	1	1	1	1	1	1	0.99	0.95	0.85	0.75	0.64
Halophosphor fluorescent	1	1	1	1	1	1	0.99	0.95	0.85	0.75	0.64
Mercury	1	1	1	1	0.99	0.98	0.97	0.95	0.92	0.88	0.84
High pressure sodium	1	1	1	1	0.99	0.98	0.96	0.94	0.92	0.89	0.85
Improved colour high pressure sodium	1	1	1	0.99	0.98	0.96	0.90	0.79	0.65	0.50	-

After initial installation, luminaire is contaminated by dirt deposition, which causes reduction of output from it. Dirt deposition rata depends on installation of luminaire, dirt type, and exposure level to surrounding environment. The luminaire maintenance factor is expressed as ratio of output of lighting device at specific time to initial output and is selected from Table 11 [17]. The environment is divided into three categories, clean, normal and dirty environment. Obviously different environment, clean interval for different type of luminaires is also different. Computer center or hospital is viewed as clean, office, shop, university or lab is considered as normal, and steelwork or chemical work is viewed as dirty environment.

Table 11. Luminaire maintenance factor in clean environments.

Luminaire type	Time between luminaire cleaning (years)					
	0.5	1.0	1.5	2.0	2.5	3.0
Bare lamp batten	0.95	0.93	0.91	0.89	0.87	0.85
Open top reflector (ventilated)	0.95	0.90	0.87	0.84	0.82	0.79
Closed top reflector (unventilated)	0.93	0.89	0.84	0.80	0.77	0.74
Enclosed (IP2X)	0.92	0.88	0.85	0.83	0.81	0.79
Dustproof (IP5X)	0.96	0.94	0.92	0.91	0.90	0.90
Indirect uplighter	0.92	0.86	0.81	0.77	0.73	0.70

As to normal environment, luminaire maintenance factor has total different set of values and those values are listed in the Table 12 [17]. This value is dependent on luminaire type and cleaning interval.

Table 12. Luminaire maintenance factor in normal environments.

Luminaire type	Time between luminaire cleaning (years)					
	0.5	1.0	1.5	2.0	2.5	3.0
Bare lamp batten	0.92	0.89	0.87	0.84	0.82	0.79
Open top reflector (ventilated)	0.91	0.86	0.83	0.80	0.76	0.74
Closed top reflector (unventilated)	0.89	0.81	0.74	0.69	0.64	0.61
Enclosed (IP2X)	0.87	0.82	0.79	0.77	0.75	0.73
Dustproof (IP5X)	0.93	0.90	0.88	0.86	0.85	0.84
Indirect uplighter	0.89	0.81	0.73	0.66	0.60	0.55

Dirt deposition contaminates room surface, which in turn causes change in illuminance from lighting device. The extent of change is influenced by extent of dirt deposition and inter-reflection to the illuminance. The inter-reflection depends on distribution of light and room index. Room index can be calculated by [17]:

$$RI = L \cdot \frac{w}{H_m \cdot (L \cdot w)} \quad (2-25)$$

Where RI is room index, L is length of room, w is width of room, H_m is mounting height.

Downward distribution such as direct luminaires has little effect on the illuminance level on the horizontal surface. But indirect lighting is highly dependent on it. The

influence of room index, luminaire type and clean interval on room surface maintenance factor can be seen from Table 13 [17].

Table 13. Room surface maintenance factor in clean environment.

Room index	Luminaire type	Interval between cleaning (years)					
		0.5	1.0	1.5	2.0	2.5	3.0
0.7	Direct	0.97	0.97	0.96	0.95	0.94	0.94
0.7	Direct/indirect	0.94	0.90	0.89	0.87	0.85	0.84
0.7	Indirect	0.90	0.85	0.83	0.81	0.77	0.75
2.5 to 5	Direct	0.98	0.98	0.97	0.96	0.96	0.96
2.5 to 5	Direct/indirect	0.95	0.92	0.90	0.89	0.87	0.86
2.5 to 5	Indirect	0.92	0.88	0.86	0.84	0.81	0.78

Room surface maintenance factor is similar to luminance maintenance factor; it will behave differently in clean environment and normal environment. The value of room surface maintenance factor is listed in Table 14 [17].

Table 14. Room surface maintenance factor in normal environment.

Room index	Luminaire type	Interval between cleaning (years)					
		0.5	1.0	1.5	2.0	2.5	3.0
0.7	Direct	0.96	0.94	0.94	0.93	0.92	0.92
0.7	Direct/indirect	0.88	0.86	0.83	0.82	0.80	0.79
0.7	Indirect	0.84	0.78	0.75	0.73	0.70	0.68
2.5 to 5	Direct	0.97	0.96	0.96	0.95	0.95	0.95
2.5 to 5	Direct/indirect	0.90	0.88	0.86	0.85	0.84	0.82
2.5 to 5	Indirect	0.87	0.82	0.79	0.77	0.74	0.72

2.4 Web Service

Web services include protocols and standards that are used to exchange XML messages between applications [18]. Web services are requested from different clients and delivery response to requesters. Web service is able to increase the efficiency of single application development and simplify the application extension for different purposes. Nowadays, the most popular methodologies to develop web service are Simple Object Access Protocol (SOAP) and Representational State Transfer (REST).

2.4.1 Three tier architecture

The classic structure of web service is three-tier component oriented architecture, which consists of presentation layer, business logic layer and data base. This architecture prescribes separation of functionality into three layers with each layer deployed into physically separate computers. It is drawn in Figure 6. Because of the independence between different layers, modification occurring within one tier does not affect others. It also supports the deployment of distributed system which leads to easily scale the whole system.

The first tier commonly is represented by web browser on client side. Browser can send HTTP request and display response. This layer offers the interface for user request. The user can enter the address which is translated into protocol and sent to remote server through HTTP protocol. The functionality of second layer is responsible for decoding user request and producing response to present information on client side. Within this layer, the business logic is executed. In the third layer, data is managed, manipulated in the database. This layer can be accessed either through business logic layer or presentation layer. Data stored in this tier is independent from business logic. Due to object relational impedance mismatch, the data is exchanged between different formats, and the action of creating, deleting, updating and searching can be achieved.

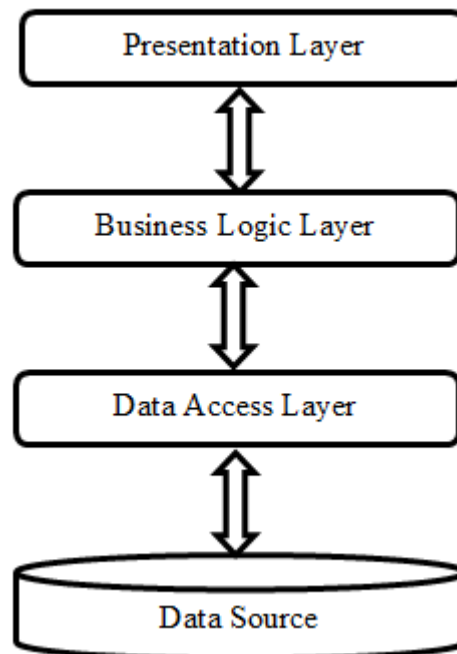


Figure 6. Structure of three tier architecture.

2.4.2 Introduction to representational state transfer

The Representational State Transfer (REST) style is an abstraction of the architectural elements within a distributed hypermedia system [19]. This provides guideline on the development of large scale web application. According to the rule from REST, web application consists of web pages which are considered as virtual state machine, and users can navigate through by links, which is stated as state transitions. Users are finally directed to the next page representing next stage of this application. REST is just a recommended architecture and it is not provided with developer toolkit. As to REST architecture, every resource is marked with URL, and service provider acknowledgements of the location of their services. All resources are categorized and accessed by HTTP method GET.

2.4.3 Introduction to simple object access protocol

SOAP is a XML based message protocol, which is used for information exchange in distributed systems by HTTP protocol. This protocol is supported by big vendors and message is easy understand due to message in the format of XML. The advantage of this protocol is that application can be deployed on different hardware platforms with different operating systems, since the exchange message is XML file. The interaction procedure is that request is initiated from client side, processed and then response is generated on the sever side. In order to achieve successful implementation, message template is defined, which regulates content, structure and order of message. Some extra property can also be defined, such as data format, and optional or mandatory. Once the message is sent or received, suitable conversion mechanism is indispensable. This is completed by XML parser, which deals with the message between request side and response side. The exchange message process is implemented by remote procedure call. The characteristic of SOAP message is that in each message, there are envelop, header and body. Envelopment element defines start and end of message, and all extra information, such as password or username, is included in header. Within body element, the actually content is specified here. The message processing procedure is that the content of head must be correctly handled, and then the message content of body part can be read.

2.4.4 Comparison between REST and SOAP

The fundamental difference between REST structure and SOAP structure is if application is resource oriented or activity oriented. REST based web service belongs to resource oriented and SOAP belongs to latter.

As to REST, resources are labelled by universal resource identifier (URI), operation performed on that resource is defined by HTTP. Essence of REST consists of four operations, GET, POST,PUT and DELETE. Operation of GET means to read content of chosen resource. Operation of POST enables to create update and delete action. Operation of PUT means that action of create and update is commended. Operation of DELETE cause resource eliminated. Point to point communication over HTTP is core

communication mechanism for REST. Communication mode is drawn in the Figure 7 [20].

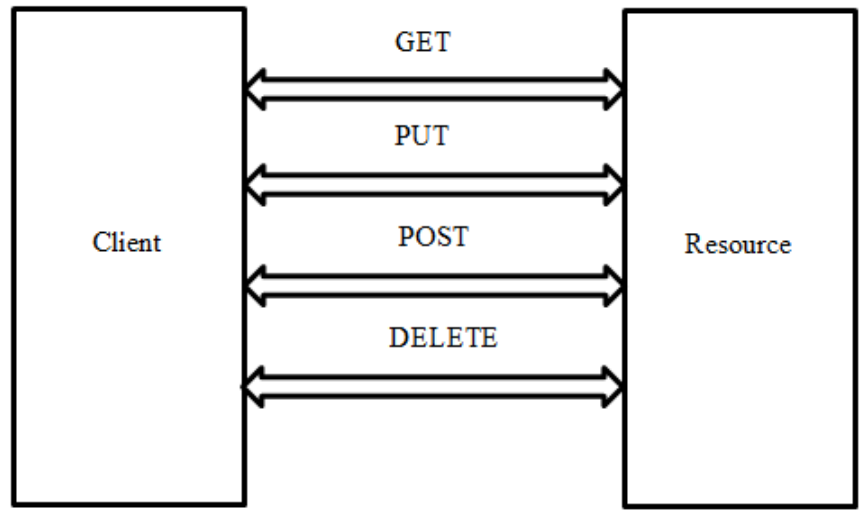


Figure 7. Communication mode of REST.

As to SOAP, action is core component within this structure, which means operation commended is relative constant, in despite of action on different resource. Communication mode is action of invoke, which is drawn as Figure 8 [20]. Implementation of action is dependent on type of activity being commended.

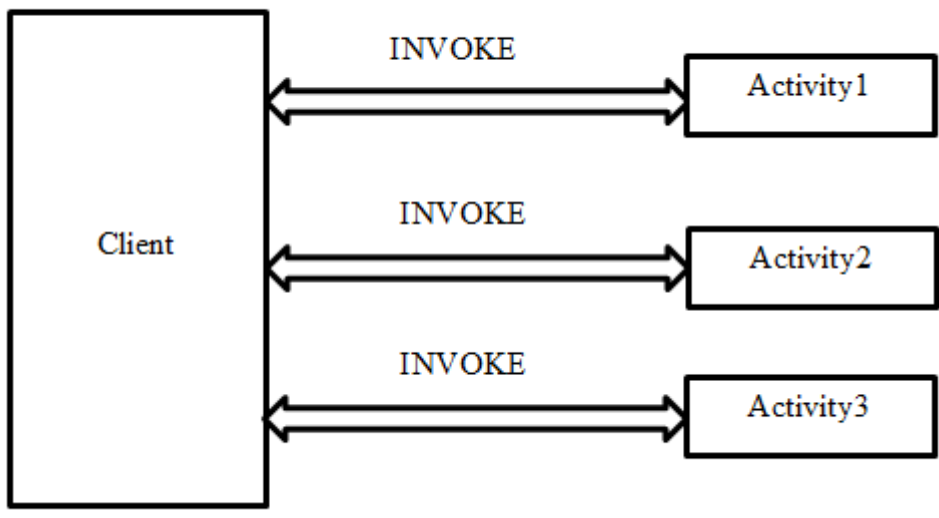


Figure 8. Communication mode of SOAP.

Due to difference between resource oriented and resource oriented structure, REST and SOAP has their own advantages and disadvantages. To make difference between them more easily to be perceived, characteristics of SOAP is listed in Table 15.

Table 15. Characteristics of simple object access protocol.

SOAP	
Advantage	Disadvantage
Structure for distributed system	Heavy weight structure
Prevailing standard	Complex message
Support by multiple vendors	More time for development of application
Extensibility	

REST is only a style of software architecture, which is developed in parallel with HTTP. This style architecture mainly consists of two components, client and server. On client side, request is first initiated, which is handled by server. Communication between client and server centers on transfer of representation of resource, which records current state of resource. When requests are outstanding, client is considered as in transition. Representation of each state includes links that can be used in the future, when client choose to initiate new transition. By implementation of representational state transfer also decouple between different services. Simplification of architecture style comes with price. It looks simpler than simple object access protocol, but it losses functionality in some key issues, such as reliability of message. Characteristics of REST are shown in Table 16.

Table 16. Characteristics of representational state transfer.

REST	
Advantages	Disadvantages
Simpler to develop	Insufficient support
Less reliance on tool	Security issue
Simple structure	Simple message
Comply with structure of Web	

In this work, the SOAP protocol is adapted. The reason is that information from message source is in the format of XML by HTTP protocol, which matches with the goal of this thesis work. Lack of support of development tool kit is also another important concern. In contract, extensibility of SOAP is highly valued, which leads to the application can be easy to be modified and integrated with new component. It is true that REST does not require XML parsing and avoid usage of message header from service provider. In this way, message content is simplified and fewer messages are compiled and transfer. Requirement on bandwidth decreases, efficiency get improvement as well. As to the complexibility of message, the structure is concise and simple, thereby it does not bring too much burden on the message handle.

2.5 Related work

Interior illumination of commercial building takes up significant proportion of energy consumption. As the environment awareness rises, the research effort to avoid unnecessary waste has been put into action by different researchers in modern time. In Luigi Martirano's research work [21], they approach this problem by comparison of two different control schema (switching and dimming) and two different actions (with or without luminaries upgrade) to analyse the cost, energy consumption and comfort. In his another published paper [22], he has studied of the power profile of lighting system with daylight and presence control. By accurately matching between the environment lighting condition and actual lighting requirement, goal of improving efficiency can be reached. In the paper published by R.Faranda, S.Guzzetti and S.Leva [23], it shows that they develop a comprehensive schema about the quality of lighting design and they use EN 15193 to estimate of the energy consumption of office building. From the research of Szczepaniak R., Wilosn M [24], They try to investigate the accuracy of prediction of the standard in the internal lighting conditions by mathematical prediction of Daysim.

3. RESEARCH METHOD

In this chapter, method of implementation of application is introduced. At the beginning, basic principles of occupancy sensor and illuminance sensor used in this thesis work are illustrated. Formation of wireless sensor network by multiple illuminance sensors and activity sensors are described. Communication between client side and server side is stated. After query data from server side, data filter processing is implemented. Data management by Hibernate framework is utilized. Comparison between different framework of web application leads to selection of suitable one. Illustration of Spring web service is also included. In the end, presentation technology used by this thesis work is discussed.

3.1 Occupancy sensor

Occupancy sensor used in this work is passive infrared occupancy sensor (PIR). The key components of this type sensor are pyroelectric detector and Fresnel lens. This detector is to inspect the change of temperature pattern across the field of view of sensor. Once change of temperature profile in the field of view is captured, pyroelectric material undergoes change in polarisation, which generates voltage signal. After that, this signal is amplified by transducer, so that change can be large enough to be measured by data acquisition system. The pyroelectric detector is most sensitive to moving objects that emit heat energy at around 10 μm , the peak wavelength of radiation coming from a human body [25]. The Fresnel lens is divided into discrete area, and when the change of temperature across some discrete area, response signal is generated. The characteristic of this sensor is that sensitivity decreases as the distance between the sensor and target object increases. The orientation and lens of the sensor determines the coverage area of sensor. The relevant coverage pattern is displayed in the figure. Most PIR sensors are sensitive to hand movements up to a distance of about 4.5 m (15 feet), arm and upper torso movements up to about 6 m (20 feet), and full body movements up to about 12 m (40 feet) [26]. The coverage of passive infrared is shown in Figure 9 [25].

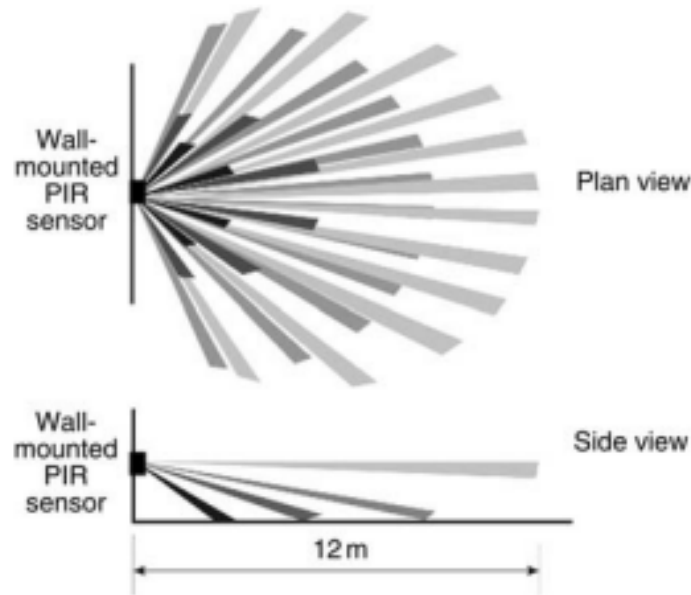


Figure 9. Typical coverage of wall-mounted passive infrared sensor.

3.2 Illuminance sensor

Illuminance sensor is used to measure illuminance level in detection area. They can measure either indoor illuminance or outdoor illuminance. Information gathered from sensor is used as reference to switch on or off lighting fixture or dim luminaires. Location of illuminance sensor is highly related to accuracy of result. It should be position at the same level of work plane. Obstacle in front of is should be removed to guarantee reliability of result. Functionality of illuminance sensor is displayed in Table 17.

Table 17. Functionality of illuminance sensor.

Category	Input / Output	Application
Indoor illuminance sensor	Input : illuminance level on work plane Output: Analogue or digital signal	visual performance energy consumption
Outdoor illuminance sensor	Input : outdoor illuminance level Output: Analogue or digital signal	energy consumption

3.3 Wireless sensor network

Traditionally, building automation system is deployed in commercial building for the purpose of monitor and control energy usage. It consists of various hardwired sensors to form a central controlled network and adapts myriad of communication protocols (LonTalk, BACnet, ModBus) . With the technology development, wireless network protocol, for instance ZigBee, HART, EnOcean, gradually replaces some of those wired links.

Wireless sensor network (WSN) is widely studied technology, which can be utilized in numerous applications of different fields, such as energy monitoring, building automation. A Wireless Sensor Network consists of several measuring devices (nodes), which autonomously form a communication network [27]. Wireless sensor can be categorized in several types. In this thesis work, only activity sensor and illuminance sensor are selected to form the monitoring sensor network. The structure of the whole network can be viewed as following Figure 10 [28].

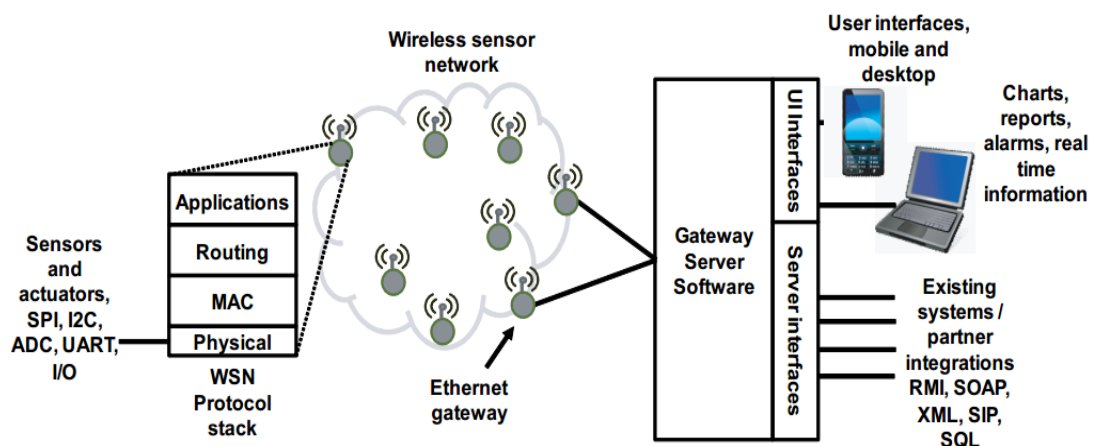


Figure 10. TUT Wireless sensor network infrastructure.

The wireless sensors will be deployed in different rooms to form sensor network. Because communication range of sensors is limited, they should be installed close to hot spot. The address of sensor is set by control panel provided by the vendor. In operation mode, sensor will monitor change of the surrounding environment and send environment data to gateway by wireless communication and then data will pass through the fire wall and transferred to the company database by Ethernet. According to transmission control protocol/internet protocol, data will eventually send back to user computer for data analysis.

In order to obtain sensor data from company, the connection between client side and server side is established first. The server runs on company computer and there is a socket which listens to request from client. On the client side, it will try to connect server with predefined server address, port number and password arranged by the company. Upon the connection built, server will create a new socket bound to the same port, so that the server can handle the request from client side, keep listening and search for new client. The client will also generate a socket by which it can start communication with server. At this time, the client can send request to server and in turn server will send out response with corresponding data back to the client. The whole communication process is illustrated in the Figure 11[29].

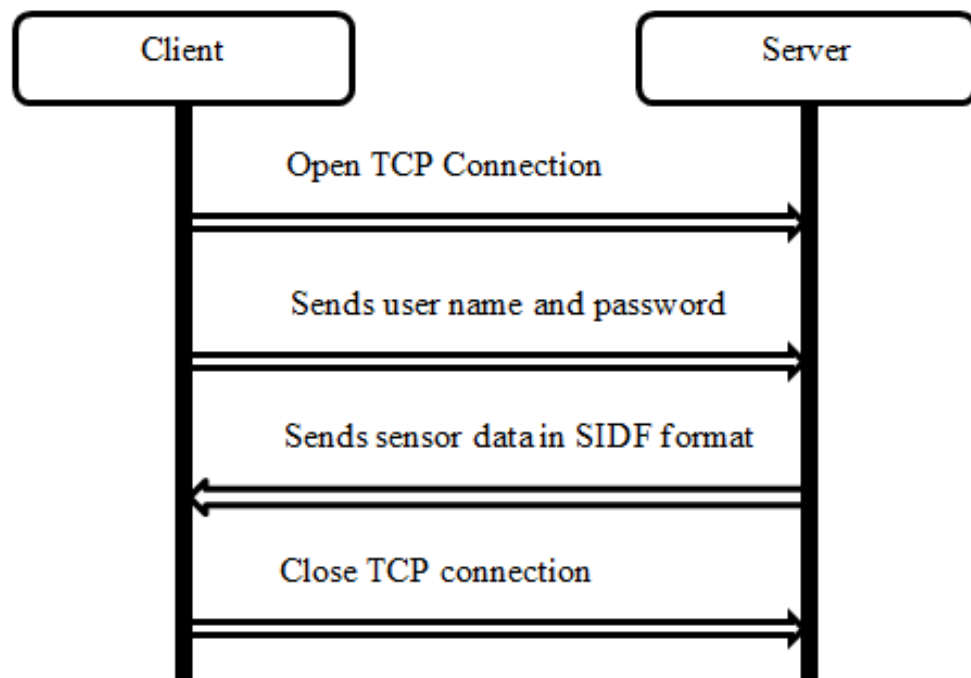


Figure 11. Client server communication sequence.

The transferred messages are in the format of XML, which is displayed in the Program 3.1. From the xml file, information about sensor node can be acknowledged, such as the network to which the sensors belong, node id, sensor type, time stamp and data value.

```

<?xml version="1.0" encoding="UTF-8"?>
<SIDF xmlns="urn:ietf:params:xml:ns:sidf" version="1.0">
  <Network id="5" name="verkko">
    <Node id="31">
      <Sensor id="1">
        <Measurement quantity="Illuminance" time="2011-09-30T06:39:08+00:00">
          <Component id="value" unit="lux">226</Component>
        </Measurement>
      </Sensor>
    </Node>
  </Network>
</SIDF>

```

Program 3.1. Format of SIDF message from sensor network.

3.4 Event Processing

Event processing is a method of tracking and analysing (processing) streams of information (data) about things that happen (events), and deriving a conclusion from them [30]. Complex event processing is to combine message from different sources, filter out data that does not comply the predefined pattern, and identify meaningful events. Normally numerous data about status of device from different sources are discrete event, that are not helpful for analysis process. Event processing can combine the discrete event with time and other information to formulate meaningful information.

3.4.1 Event processing engine

Event stream processing handles information from multiple streams to identify meaningful events with almost no latency, which is illustrated in Figure 12 [31]. Event processing engines is critical component of event processing platforms, which is used to deal with numerous events. Different event is handled by separate thread, and processing engine is capable of deciding which rule is applied to. Pipeline technology is implemented to guarantee that handle instructions in parallel to maintain huge throughput.

Event processing engine is defined by event processing language (EPL). EPL determines instruction that search for and acts on pattern according to inbound event stream. Instruction compiled by EPL is concise, but logic implemented by event processing platform is sophisticated. Event processing engine is the new way to process information, since data is dynamic and event processing engine acts on all the inbound message. This feature enable event processing engine can handle real time message, which is very important for business.

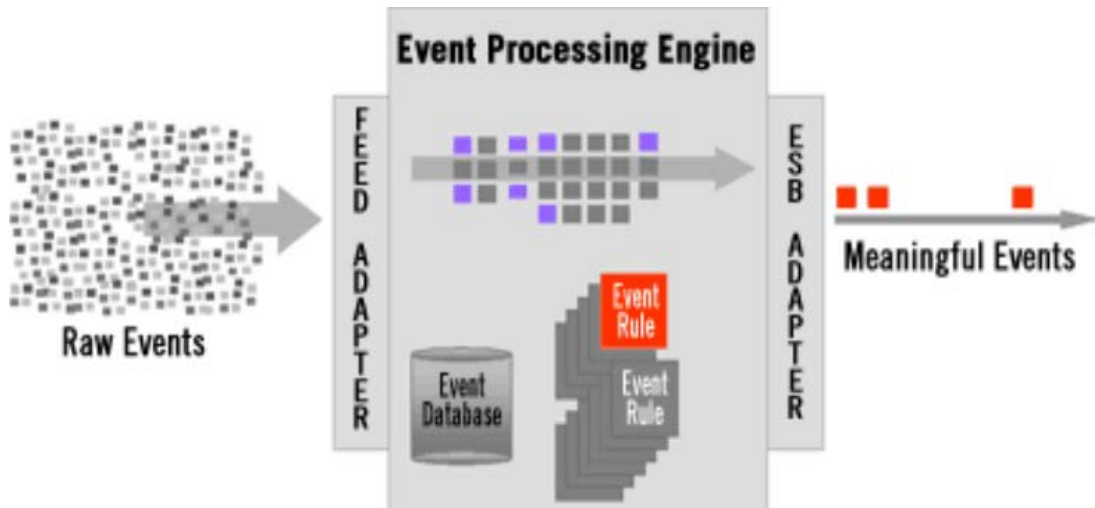


Figure 12. Principle of event processing engines.

Event processing is associated with database with different purpose. Normally, reference or threshold value is stored inside database, event processing engine keeps comparing between real time values and reference one. As long as requirement is satisfied, correspondent action is generated. The other way is to store useful information for future usage. Qualified information passed through of search engine can be considered as target information. At this moment, quantity of information is smaller than previous. Burden of interaction with database decreases and efficiency improved.

3.4.2 Introduction to implementation of event processing engine

Esper is an Event Stream Processing (ESP) and event correlation engine (CEP, Complex Event Processing) [32]. It bases on real time event driven architecture and is used to trigger action specified in plain old java objet (POJO), if particular requirement is fulfilled. There are two methods to handle high volume events. One is to store event inside database and then use customized query clause to extract target data, the other way is to filter out useless event and only concentrate on the target event by defining search criteria. The principle of Esper bases on the second method, the events stored in the database are the only relevant event. In this way, Esper is perfect alternative to handle real time high volume event. It enables application to store query clauses and monitor income data instead of store every data at first and then run though query criteria, so numerous irrelevant events are not stored in the database which lower the burden of database. It also provides solution to real time query.

Esper can register query criteria in the search engine and define listeners which will be invoked as long as the condition of query criteria is satisfied. The query condition can be written according to temporal window, filter, aggregation, and sorting. To define application of Esper, determination of certain type of qualified event and irrelevant event is the first step. To improve the efficiency, throughput and latency need to be

defined. Information source, format, content and relationship between unit events are used to define event stream. Specification of representation of event format and EPL statement is the last step to complete building up Esper application. There are two ways to define pattern detection, one is EPL pattern statement, the other is match recognize. The comparison of those two statements is listed as Table 18.

Table 18. Comparison of EPL statement and match-recognize statements.

Category	EPL	Match recognize
Scope	Sequence of events	Sequence of events
Real-time application	Real-time	Real-time
Event type	Monitoring different types of event	Single type event
Data window	Pattern state is independent to leaving event	Pattern state relates with leaving event
Evaluation	Truth-false value based	Sequence based

3.5 Data base connection

When user application accesses to data, data base connection is first built up, which allow communication between client software and database server software. Location of database server software is arbitrary, which can either on the same computer with client application or not. After connection is completed, commend and answer is transferred between them. Each action relative to data base requires connection, so connection pooling is used to improved performance. In connection pool, predefine number of connection is stored. When user send out commend to fetch data, connection pooling will automatically assign one session to it. When the second requirement arrives, the other session is built. The number of session available to connection is limited. If number of requirement are larger than number of connection within connection pool, this requirement is on hold. When action is completed, session is released and number of session in connection pool will increase. Normally, open and close of database connection require numerous extra codes to implement and those codes are almost identical which leads to unnecessary repetitive work. To tackle with this situation, database management application is given to birth. Those softwares are developed by big company which will take care of routine work. It also offers extra feature such as improvement of efficiency. The two most famous framework are Java database connectivity and Hibernate framework which are described as below.

3.5.1 Java database connectivity

JDBC is Java based data access solution. JDBC API is used to build up database independent connectivity between Java programming and various databases. Database connection is very important for the application. Without connection to the database, web application only displays static content. Introduction of database connection provides dynamic data flow to the application. The most common action of database consists of creating, updating, deleting, and reading. Those functions are highly supported by JDBC, which provides specific functions to be called. Within JDBC, there is also pre-processed mechanism available for user, so that the action of database can also be dynamic. Query is dynamically assembled during proves. With the help of JDBC, process of accessing data base becomes simplified. It consists of two different sets of interfaces, one is for application writer, and the other is for driver writer. JDBC driver can enable the application access to data base by JDBC API. Driver translates JDBC call into network protocol by DBMS so that client machine can directly connect to DBMS server. It is also feasible that JDBC call is translated into middleware protocol, which then converted into DBMS protocol by middleware server, so the connectivity to various different databases is achieved.

JDBC API loads Java package and registers it with JDBC driver manager, which is used to create JDBC connection and acts as connection factory. The action command can be written in the format of statement. For the purpose of reusability, prepared statement is adapted, since it is cached and can be execute repeatedly. The whole structure of JDBC is described in the Figure 13 [33].

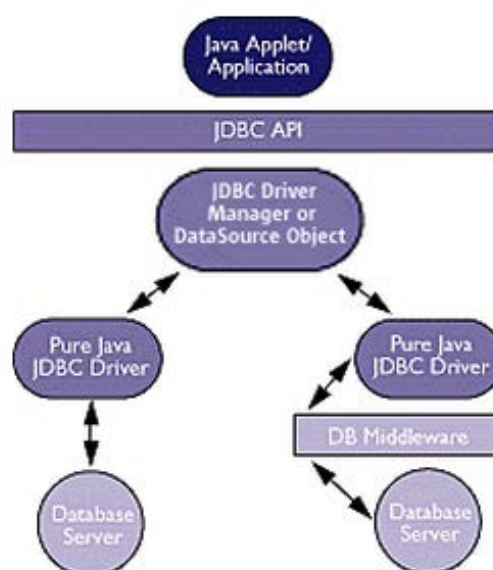


Figure 13. Description of JDBC.

3.5.2 Hibernate framework

Hibernate framework is an object relational mapping library for Java. It provides solution to conversion between object in Java and relational entities in database, which is required in most of web application development. The work procedure becomes standard which dramatically save lines of code and reduce development period. To set up framework, database is built up at first. Correspondent of plain old Java object (POJO) in database is defined and the property of POJO which requires being persisted is indicated.

The Hibernate mapping file, which is responsible for mapping the properties of POJO to columns in the table, is created.

Hibernate libraries is added to the class path of application and Hibernate configuration file that connects between database and mapping file is created. JDBC driver is also included in the class path, since they are independent to each other. The dialect of relational database that abstracts the difference between different relational database is chosen

MBeans which are the wrappers for resources and services are utilized for managing application. `HibernateServiceMbean` is the interface within the package of `org.hibernate.jmx` and `HibernateService` implements it. The getter and setter method for data source, transaction and dialect is defined by it. When `HibernateServiceMBean` executes, it creates configuration object from mapping file and `SessionFactory` object. `SessionFactory` is defined to JNDI location based on `JMXMBean`.

Within the application code, Hibernate configuration object that refers to configuration file is created. The `Configuration` class is responsible for reading configuration information, and configuration object is used to build up `SessionFactory` object. Only one `SessionFactory` exists in the whole life cycle of the application. From `SessionFactory`, `Session` objects are created whenever application interacts with database. Within one session, action of adding, deleting, updating and querying of object can be executed, and result is persisted. After action is completed, session is closed.

The alternative connection between mapping information property and hibernate application is defined by configuration file, `hibernate.cfg.xml`, so that within application the, relationship between java classes and database table is informed.

The Hibernate `Session` can be extracted from `SessionFactory` and the codes are wired together so that functionality of Hibernate can be accessed from Java application. The Hibernate plays the role as data access layer. The advantage of Hibernate can be further exploited by the support of Spring framework, which offers support for persistence object

and transaction management. The general view of Hibernate framework is drawn in Figure 14.

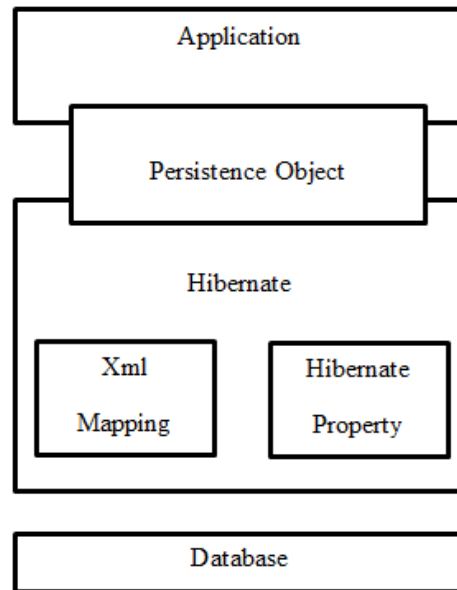


Figure 14. Hibernate work principle.

3.6 Web application structure analysis

To build up we application, there are lots of different framework available to choose. Some of them bases on same principle with different variety, some of them bases on different concept. In this section, different frameworks and principle behind them are reviewed and compared. In the end, it explains the reason about selection of web application framework.

3.6.1 Model-view-control structure

The model, view and controller (MVC) architecture divides application into three separate domains: model, view and controller and this structure is shown in Figure 15 [34]. This structure provides extra flexibility to application, so that it can be fast adapted to the new requirement. Interconnection between those three components is also depicted. User initiate commend to controller, which handle this commend by change of state in model. After that, controller obtains new value of model and select view component to send back to user. On user side, state query and change notification are perceived. In the figure, event is illustrated by dot line, which is perceived by end user. Solid line is used to represent method invocation, which is the mechanism of actual logic procedure. This structure replaced with traditional three layers architecture by its explicit logic process.

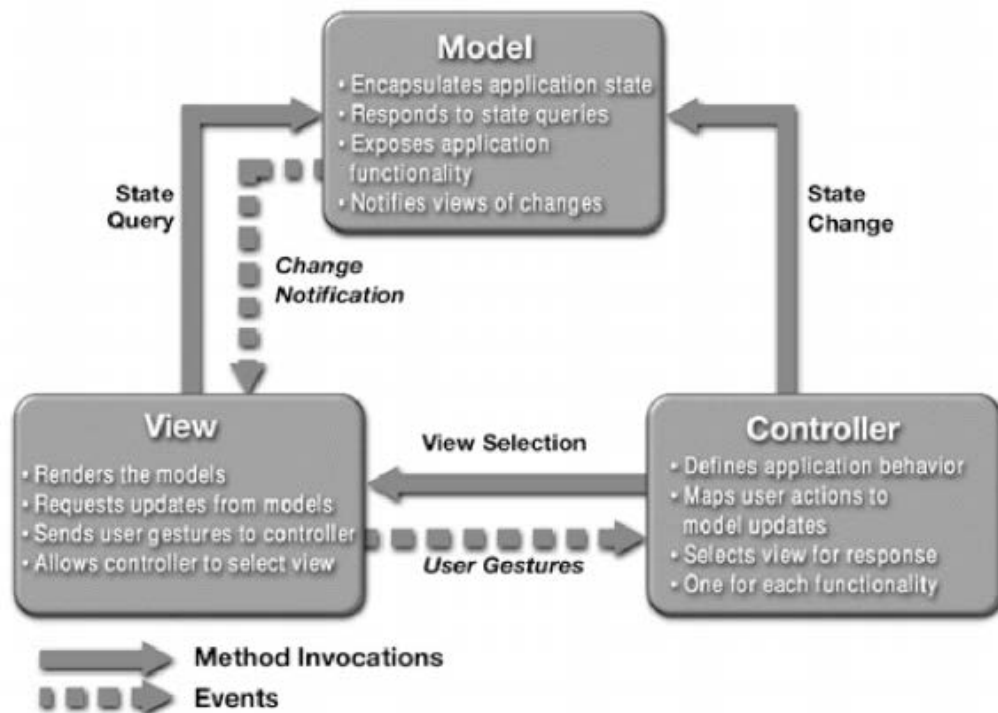


Figure 15. Model-View-Controller model.

The model represents data of application and corresponding method that manipulates data. Data is part of persistent state of application should contain within model objects. The model provides generic service to variety of clients and the combination of the data and action can provide specific service. The nomination of each service is designed to be self-explain for the sake of convenience. Those services are abstracted and will be overwrite according to business logic. The interface exposes methods to higher level for modification of property of model or implementation of complex business logic. The view layer will be notified if the value of model has updated.

The view is used to either render the state of model or forward user input to controller. In MVC structure, view layer and model layer are separated so that model does not have dependence on view layer.

The controller is responsible for translating the user input into action and being directed to model. It is also responsible for selecting appropriate view layer so as to display the result.

The work principle of MVC structure is described as follows. The user interacts with user interface and controller handles input from user interface by registered handler. The controller accesses to model and processing business logic. The properties of model are updated and redirect to view layer. The view layer will extract data from model and send result back.

3.6.2 Struts web application

The Struts is a reusable half-completed framework that can be customized for specific requirement and structure of Struts framework is displayed in Figure 16 [35]. On the client side, the HTTP request is sent to the server and the work flow is controlled by ActionServlet. When the request from container is read by ActionServlet, it uses universal resource identifier (URI) to determine which Action is used to process request. Action does not only have ability to access to business layer and retrieve information from database, but it also is responsible for validate input. The input data from request is stored into JavaBean by ActionServlet which are subclass of ActionForm. The bundle between ActionForm and input data is determined by ActionServlet. After those data manipulation, result can be forward to another resource by Action or the logical name of path to other page, which is stored in the ActionForward. When business logic is executed, ActionForward returns to servlet by Action. The target page can be found according to the path store in ActionForward by servlet to complete the response process. The whole process is included in the ActionMapping which is defined by specific path. When request from client is received, servlet will read the ActionMapping which determines specific Action, ActionForm and ActionForward to use. The configurations of Action, ActionForm, ActionForward and ActionMapping are stored inside the struts-config.xml. At start up stage, Servlet reads this file and creates database configuration file.

Model of Struts consists of two components, internal system state and business logic work against state. Internal system state is represented by ActionForm bean. ActionForm is used to preserve parameters from user request. Property name and HTTP request parameters are the same as the name of controller parameters. If it is necessary, those information are stored into database. It will be easier if property names and types of bean are the same as names and types of column of data table.

In Struts framework, it has its own controller which can be integrated with model and view from other technology. As to the model, Struts can access to standard data access technology such as JDBC, EJB, and Hibernate framework. As to view component, Struts is capable of interaction with JSP, Velocity template. Support to other technology provides Struts extra flexibility, when it is adapted to develop web application.

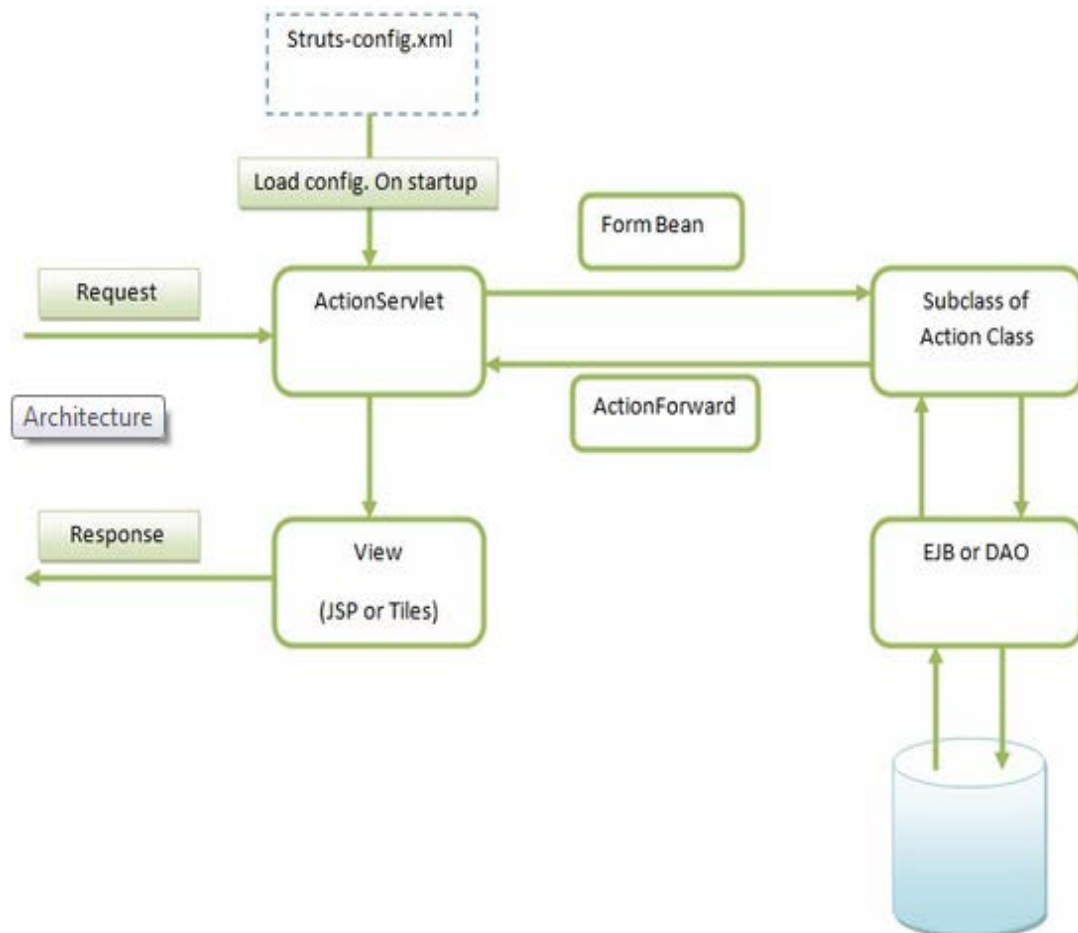


Figure 16. Working procedure of Struts.

3.6.3 Struts 2 web application

Struts 2 framework is originated from Struts 1 framework and integrated WebWork as well to form a more flexible front controller based web development paradigm. To facilitate the development of project, Maven can be integrated to work with. Maven is a build automation tool which can manage project written either in Java or C#. It provides one powerful and handy function that is automatic support package import. Struts 2 framework normally requires extra support from different sources. Identification of each sources slow the development process. With the help of Maven, all the required and latest version of package is imported into project, as long as configuration file is completed.

The whole procedure is described in Figure 17 [36]. On client side, user initiates request to server side. On the server side, FilterDispatcher receives request and redirect request according to predefined rule. On the way to execution of action from FilterDispatcher, there are multiple interceptors in the way. Their main task is to apply necessary function on the message flow including validation user information and file upload. It is automatically trigger, when message flow pass through. Upon message is arrived at action, business logic starts to implement. Communication between application and database is triggered. Connection channel is built up, information exchange and data manipulation is performed. The most common action contains creating, deleting,

updating and adding. This result is stored into model and send back to FilterDispatcher. The same thing occurs again. Interceptor will perform action on back flow of message again with the same condition, but in the reverse order. Finally, control is return to container, which sends response back to user side. The whole communication bases on HTTP. Request from user is represented by HttpServletRequest and response is indicated by HttpServletResponse. Struts.xml is the main configuration file to determine whole application.

The useful feature of Struts 2 is that inflow message can trigger chains of interceptors. When the message arrived, front controller first determines Action class to handle with this request and interceptor from action mapping. Instance of ActionInvocation is created and invoke method is called. In this stage ActionInvocatoin take over control from front controller to be in charge of further process. Combination of Interceptor and Action is implemented by ActionInnvocation which also defines invocation sequence. First invoke method from stacks of Interceptor is invoked and action is performed against income message.

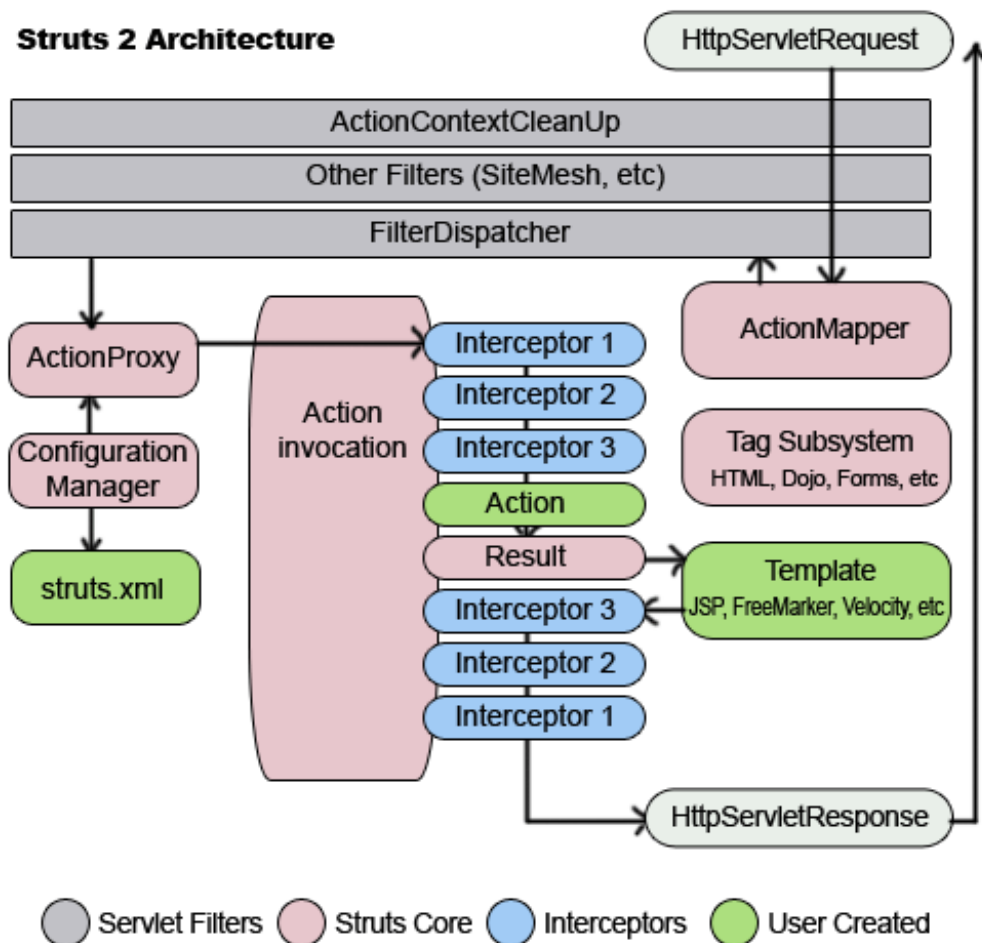


Figure 17. Architecture of Struts2.

3.6.4 Spring model-view-control

The main component of Spring MVC is DispatcherServlet that dispatches requests from client side to appropriate handlers by configuration of handler mappings, view resolution, theme and locale resolution. It is designed to handle every web request and determine the handling process, which acts as the front controller in the design pattern of JAVA EE. Then it will configure different components defined in web application context to solve the request. The definition of controller can be achieved by annotation. The handler method in the controller class is marked by @RequestMapping and the request which goes through controller can be directed to the right handle method by handler mapping. Once the choices of handler method are picked up, those methods are executed according to business logic. The business logic interacts with lower level method to process request and return result back to high level. It reaches business object which is the combination of granularity services to provide comprehensive service. Then it is directed to DAO layer, in which the action of creation, modification, update, and delete is defined. Once the process flow goes back, the controller dispatches the flow to view layer based on the return value from handler method. The value from handler method does not specifically indicate the view implementation, that is only logical name is returned, instead of concrete name.

To pass the data from controller to view layer, the transfer object is the input argument and the logical view name is translated into concrete view name by view resolve. View resolve is defined in the web application context, which is responsible for matching between abstract view name and concrete one. This whole process is explained in Figure 18 [37].

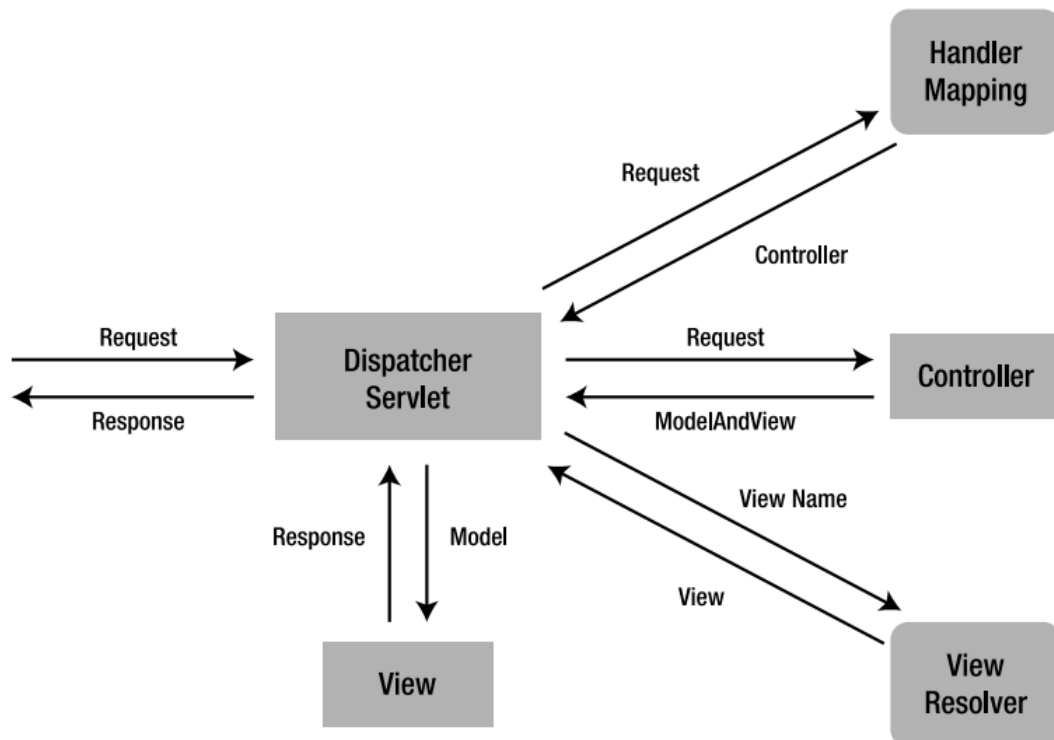


Figure 18. Work principle of Spring MVC.

3.6.5 Web application structure comparison

From Struts 1 to Struts 2 framework, structure and principle of framework have huge difference. This huge difference can be derived from comparison Table 19. To interact with user input, Struts1 uses ActionForm objects to capture information. The format of user input may be different from property of ActionForm, so converter is required. In Struts 1, Commons-Beanutils is adapted. After data conversion, manual validation or object from Commons Validator is used to verify the input information. Action classes which extend abstract base class process input and delivery result to view layer. Only one instance of class exists and handle request. All action within same module share same lifecycle. The result binding with view is achieved by JSP mechanism. As to Struts 2, ActionForm object is replaced with Action properties to obtain user input and it is converted to specific type by OGNL. Validation process is completed by manual validation or XWork validation. Action class is implemented from Action interface, instead of extension of abstract base class. Multiple instances of Action class are created correspondent to each request. Each of different Action has their own separate lifecycle. Separation of view and object type is achieved by value stack technology. View which has same property name but different property type can be reused.

Table 19. Comparison between Struts1 and Struts2 framework.

Category	Struts1	Struts2
Input binding	ActionForm object or DynaBean is used to capture input	Input property is transferred to property of Action object
Conversion mechanism	Commons-Beanutils for type conversion	OGNL for type conversion
Validation	Manual validation and extension to Commons Validator	Manual validation and XWork validation framework
Action class	Extension of abstract base class	Implementation of interface
Thread model	Singleton design	Multiple instances
Action Execution	Separate life cycle for each module and all Action in the module share same lifecycle	Action based lifecycle.
View layer binding	JSP mechanism	Value Stack

Spring and Struts2 framework share lots of features in common, but difference still exists. Comparison between Struts 2 and Spring framework is illustrated in Table 20. They all bases on the principle of MVC structure, model, view and controller layer are separated. Struts 2 provides many complex tags for building interface, while Spring framework only offers some basic HTML tags. So Struts2 are more easily to develop consistent pages. Struts2 is presentation layer framework, which is mainly used to

navigate among pages, and has integrated validation function. Spring is more than just presentation layer framework, it is a complete J2EE framework and can be integrated with other framework. It also provides support for transaction management of database.

Table 20. Comparison between Struts 2 and Spring framework.

Category	Struts 2	Spring
Popularity	The most popular framework	Very popular framework
Presentation support	It offers many complex UI components	It only provides basic HTML tags
Framework structure	Presentation layer technology	Complete J2EE framework
Transaction management	No support	Fully support

3.6.6 Integration of Spring framework with Hibernate framework

Hibernate framework offers the ability of persisting data in database, and Spring framework provides the capability of IOC. The combination of those two technologies can provide more powerful function to the application. Hibernate mainly is responsible for managing data exchange between application and database, which provide consistent platform independent manner to access various databases, since low level process is abstracted. Hibernate is truly helpful for data management and persistence, but the manual management of its objects, such as SessionFactory, and Session brings new trouble, since they are distributed across different parts of application. With the help of Spring framework, all those objects are registered in IOC container and the property of objects can be easily configured.

The integration of Spring and Hibernate starts with creating database and target object whose property name is the same as the column of data base. The mapping file, which describes the mapping from relational database to the JAVA object, is created. The XML based Spring configuration file that is responsible for configuring business objects managed by IOC is created. Within this file, important properties are specified, such as data source, SessionFactory, Hibernate Template, and DAO class. The data source is used to determine the connection properties that are needed for utilizing the database. As to the session factory, the data source is referred and the mapping file is included. Hibernate Template whose property refers to Session Factory contains low level yet important functions, such as creating, deleting, updating and querying. DAO object which depends on either Hibernate Template object or Hibernate API is used to provide various services for the client. After all those configuration and design is completed, application can take advantage of it. It will fetch DAO object from the configuration file; Hibernate Template object is initiated, and SessionFactory is referred. At the same time, data source is resolved.

3.7 Spring Web Service

There are two different way to create web service, one is contract last and the other is contract first. As to contract last, Java code is first developed and web service contract is derived from that. As to contract first, web service contract is written by web service description language at first, then use Java to implement it. Spring web service framework developed bases on contract first principle.

3.7.1 Web service definition language

Web service definition language (WSDL) is an extensible mark up language (XML) format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information [38]. There are several important elements in this standard, types, message, operation, port type, binding, port and service.

Type defines data format for exchange data. For the purpose of high interoperability, XSD is adapted to define message type.

Message consists of group of parts. Part comprises of element and type. Element refers to XSD element. Type refers to simple type or complex type defined in previous to regulate and verify data format. It only describes abstract content of message.

Port Types is a group of abstract operations and abstract messages. There are four different types of operations, one-way, request-response, solicit-response and notification. One way indicates mode that endpoint receive messages. Request-response means that endpoint receive message first and sent response back. Solicit-response means that endpoint initiate message sending and receives response. Notification means endpoint only sends out message.

Feature of bindings is to assign operation and message of port type with concrete message format and protocol.

Function of ports is to define individual endpoint with unique single address, such that this resource can be identified by application.

Service is group of related ports. Capability of resource is exposed to others by distribution of description of service. Different operation within service is synergized to complete task.

In Figure 19 [39], structure of WSDL is illustrated and difference between WSDL 1.1 and WSDL 2.0 is shown. On the abstract level, types are assigned to individual message to define data format and message is referred by input and output of operation. Operation is individual action that is responsible for implementation of certain action on abstract level. Port type consists of group of operations which are assign concrete message format and communication protocol by binding process. In the end, service of this source is exposed to others.

In version 2, structure of WSDL is simplified and in line with notation of programming as well. Name of port type and port used in previous version are switched to interface and endpoint, which leads to easily understand. Types in version two combines notation of types and message in previous as well.

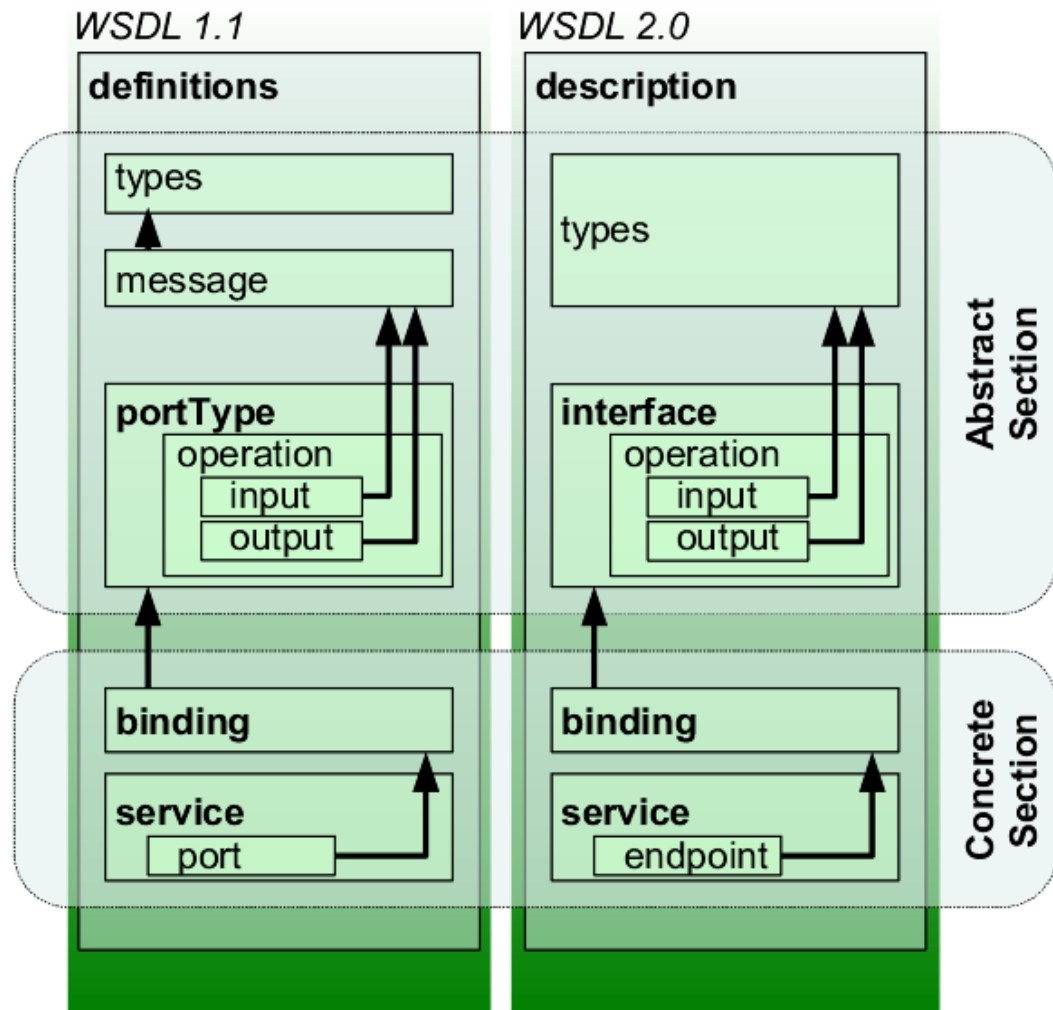


Figure 19. Comparison between WSDL 1.1 and WSDL 2.0.

3.7.2 Java architecture for XML binding

The combination of XML and Java offers the possibility of exchange data across different platforms, since the XML is the standard for conveying message across distributed system and Java language provides platform for portable application. JAXB provides services for accessing and processing XML data with minimum effort. As to the server side, XML based various data are sent from field devices and the conversion from XML message to correspondent JAVA object with the same content and structure is completed by JAXB. This process is called unmarshalling. After unmarshalling XML document, application can access and display it by reading from JAVA object. The process of unmarshalling consists of two steps, binding schema and unmarshalling document. Schema is the XML specification which regulates content, order and structure of element. The result of schema binding is to generate set of JAVA classes and interfaces that reflects the schema by binding compiler. The getter and setter method within the classes are used to define the property of element and attribute. After unmarshalling, java object that represents the XML message is created, which is depicted in Figure 20 [40].

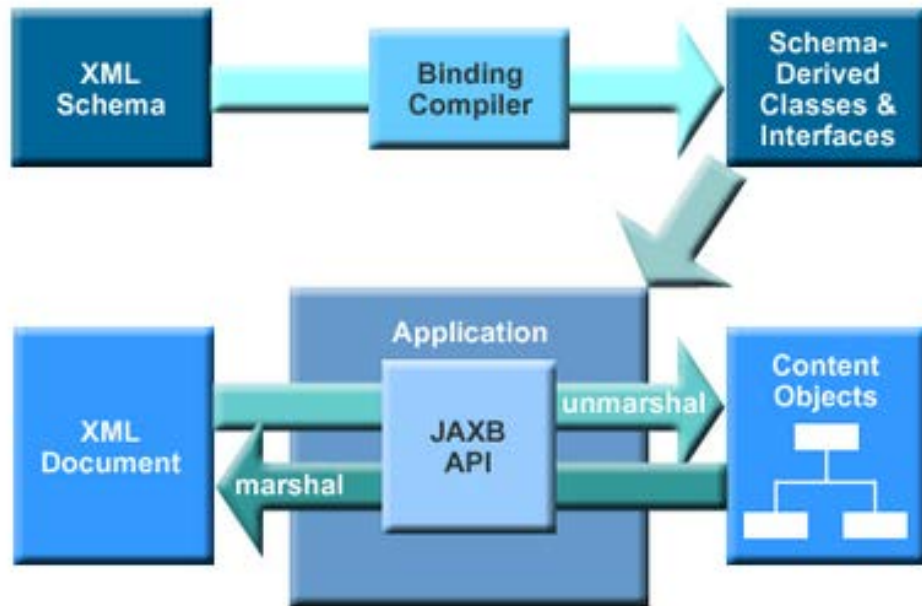


Figure 20. Basic principle of JAXB.

3.7.3 Inversion of control

Ioc stands for inversion of control, which consist of mainly three components, object, configuration metadata and Spring container and this shown in Figure 21 [41]. Their relationship is depicted as following picture. Spring container requires configuration metadata as input, which determine configuration of Java bean. Instantiation, configuration and assemble is completed in this stage. There are lots of different ways to define configuration metadata. The most convenient way is to provide XML file. Container is the place that action of management of Java bean takes place. Bean Factory is used to represent container. The main task of it includes instantiating and configuring object and injecting of dependence into object.

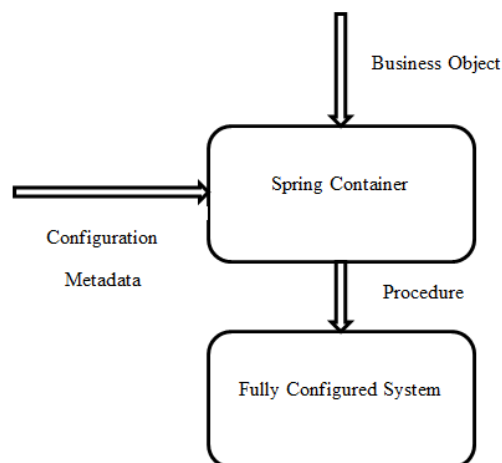


Figure 21. Principle of IOC.

Normally, web application consists of more than one object to form required function. Coordination between different object is essential for the success of application. Introduction of dependency injection is used to bundle stand-alone object into loose-coupled entity. The dependency can be formed by constructor argument or property of object.

3.7.4 Spring web service

Spring Web Service is used to facilitate contract-first, document-driven SOAP service development. It provides powerful mapping ability, because incoming message can be distributed to any object according to message payload or Action header. The process of conversion of incoming message is simplified by using marshalling technology, such as JAXB, instead of DOM, SAX, or JDOM.

There are several main components defined in the Spring Web service on server side, such as MessageDispatcher, and Endpoint. MessageDispatcher, which dispatches all the incoming messages passing through interception and generates response from different sources to appropriate endpoints, by the content of endpoint mapping. The capability of MessageDispatcher is the combination of DispatcherServlet and MessageDispatcher, which is configured in the web.xml file to delegate the mapping relationship. Endpoint is marked with @Endpoint annotation, which contains set of various handling methods so as to extract part of message and generate response. The request handling process can be described as follows. Central MessageDispatcher is set up and request for this specific dispatcher comes in. Appropriate Endpoint is pinpointed by the content of EndpointMapping and the request goes through pre-processors, endpoint and post-processors. Endpoint is invoked by the defined EndpointAdapter. Important components and interaction process is shown in Figure 22 [42].

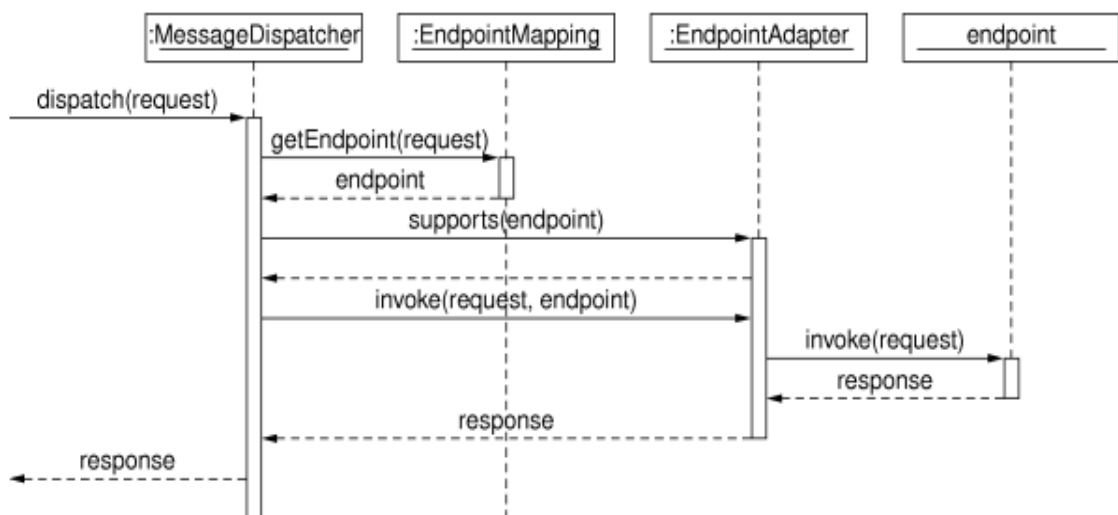


Figure 22. Message flow in Spring Web Service.

As to the information transportation from client side to server side, automatic WSDL exposure is taken into use. It has two different method, static exposure and dynamic

exposure. Static exposure means the WSDL description is written in WsdlDefinition beans and it is detected by MessageDispatcherServlet and exposed to client by WsdlDefinitionHandleAdapter.

3.8 Presentation technology

Interface of web application plays two roles, one is to capture data from user, and the other is to present result to user. To make data easier to be understood, visualization of data is necessary. In this section, different presentation used in this thesis work is evaluated.

3.8.1 Java server pages

JSP stands for JAVA Server Pages, which provides method to create dynamically generated web page based on HTML or XML. It processes similar functionality of servlet and is translated into servlets at runtime, so it can be considered as abstraction correspondent of it. JSP is component of model-view-controller design which can interact with java bean and delivery final result back to the client side. User interface is separated from content generation by JSP so that modification of page layout does not necessary change the dynamic content at same time. Both Scriptlet and JavaServer Pages Standard Tag Library expression languages are the two alternatives for the choices of JSP programming language. They look like XML-like tag, and the business logic is written within it for generating web page. All those tags are sent back to the response page for displaying.

3.8.2 jQuery

jQuery is a JavaScript library designed to simplify the development process and reduce programming period of client side web page. It offers very useful code library and dramatically reduce the coding task. The process of selection of DOM elements, document navigation, animation effect and event handling become concise and high efficiency. To locate desired target within DOM file does not require go through whole structure step by step, instead the appropriate combination of selectors can easily achieve the goal. The ready-made animation function can satisfy common requirement, so that duplication work load is avoided.

To improve the interactivity of dynamic generated web page and reduce the response time, AJAX technology is adapted. People are used to quick response and interactive control during usage of desktop application. However, it is total different story for web application due to different mechanisms. Web application is usually slower than its counterpart of desktop application, since it only works on browser and communication between client and server require extra time to process. Normally, user send request to service, which contains lots of information. After initiation of request, users loss control over its application, and waits for reply from server. On server side, it requires to receive request and start to generate correspondent response and send back. Double

travel time can compromise usability perceived by user. Even worse, when the traffic on that web site is heavy, response is painful slowly. Ajax is given birth to tackle with this troublesome issue.

AJAX is a programming architecture to generate highly interactive web application. It is done by sending out only partial web page instead of whole page. This functionality is implemented by JavaScript routine as middle man and HTTP request with XML data. This avoid situation that entire page need to be sent back and forth between server and client. This whole procedure can be viewed in Figure 23 [43]. Request sent from client does not directly go to server side; instead AJAX routines receive JavaScript call. Then it forward requested data to server and get response from it in the format of XML. Upon it receive response; it automatically updates content of webpage. Advantage of this technology is that fewer and smaller calls are sent to server, more requests are processed, handled locally. This reduces traffic burden and fast response.

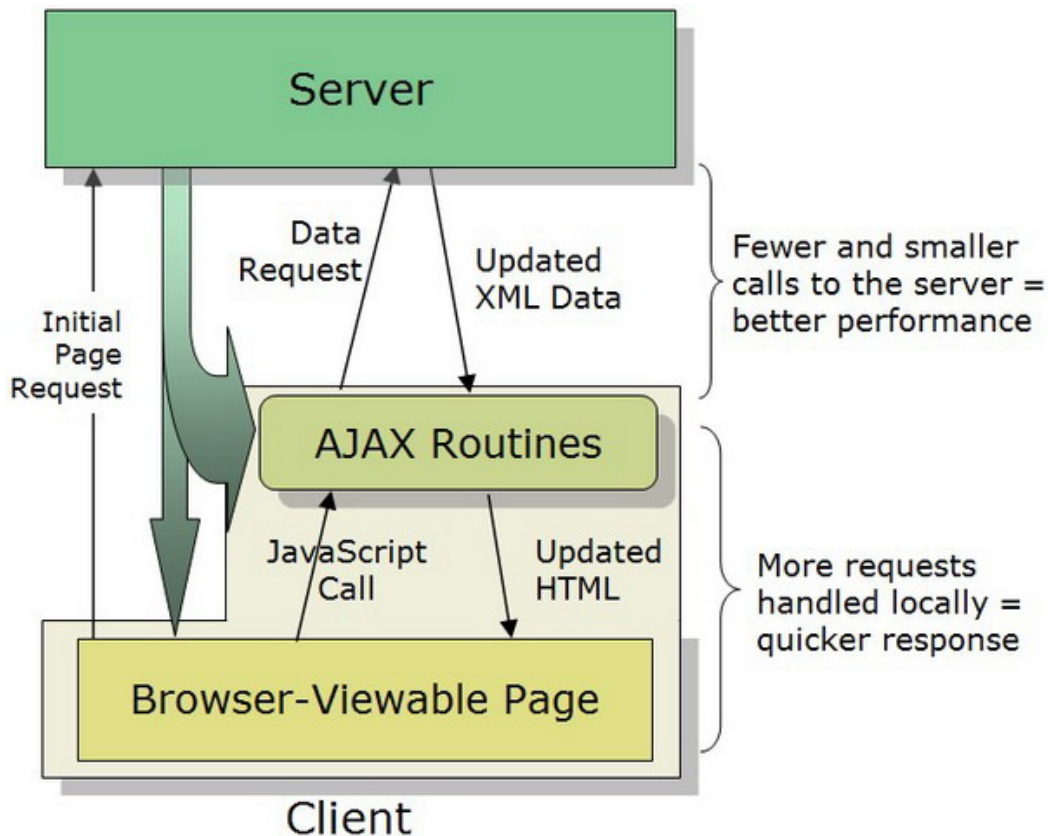


Figure 23 Principle of Ajax.

3.8.3 Tiles

Web application consists of pages that have different independent pieces and similar overall structure, which is shown in Figure 24 [44]. Web pages with similar structure and different content of each section in various situations can be designed by implementation of Tiles. It consists of three main parts: template, attribute and definition. Template defines the layout of common page. Attribute is the component within template that needs

to be filled in the application. After combination of template and attribute, definition or abstract definition is produced and rendered to the final user. The mechanism of tile is to develop unified template for different web pages and includes different section within this framework, which offers flexibility. Normally, content of some section of framework is the same, but others change. If web pages do not require any modification, they are just integrated without any effort. If new content are updated, the same change will be displayed in the framework accordingly.

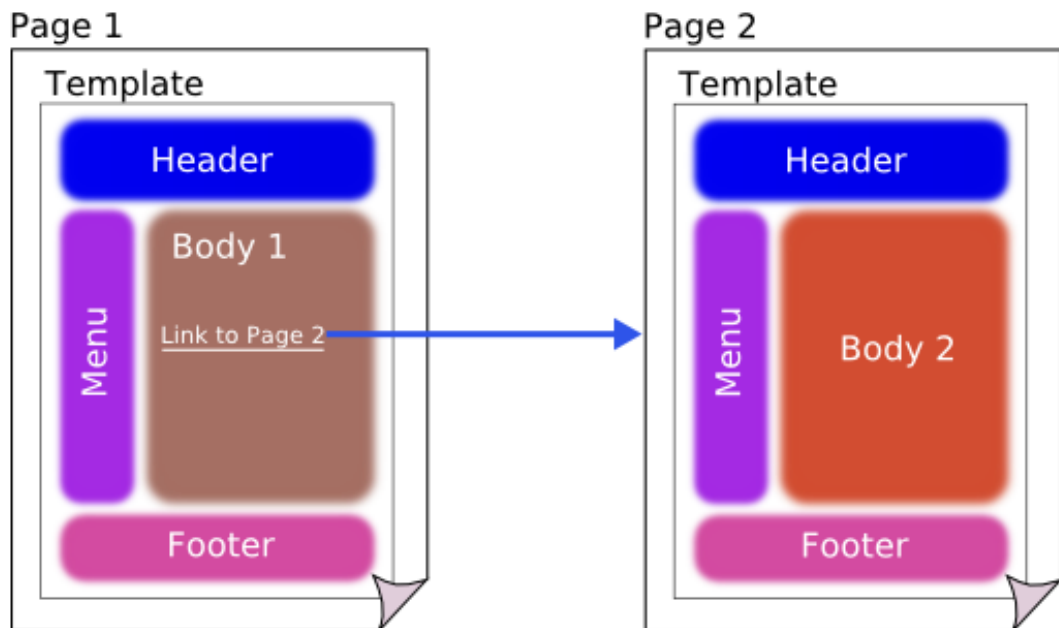


Figure 24. Principle of Tiles.

3.8.4 Google chart

Google chart is responsible for visualization of raw data on webpage. It provides various chart formats, from line chart to tree chart. It is divided into three sections: data base, data table and customized Google chart. The environment data from wireless sensor network is stored in the database and the business layer of spring MVC accesses it and manipulates the data. Data table is responsible for converting the data into defined format and send to the chart. Different raw data is stored in data table with the same structure so that it is compatible with any raw data. The value of data table is derived from database and it accesses to the business layer of spring MVC. In chart library, different charts are exposed as JavaScript classes, and it can be easily displayed on webpage. Basic principle and main components are illustrated in Figure 25 [45].

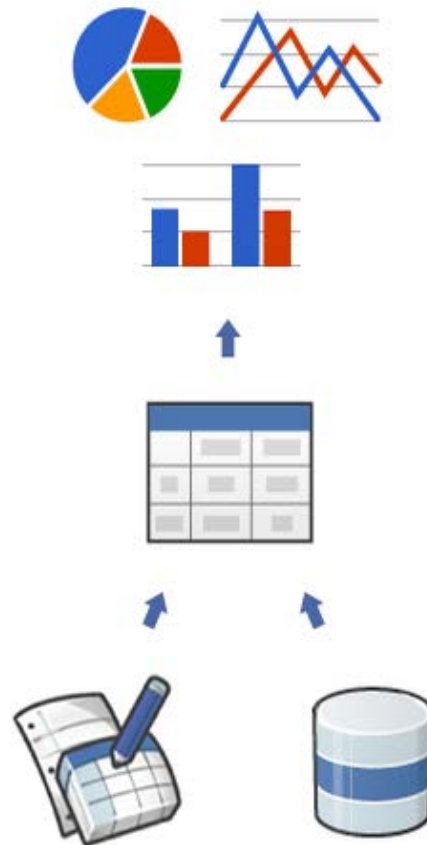


Figure 25. Principle of Google Chart.

3.8.5 Google Map

Google map is free application provided by Google. Not only it offers services such as map orientation, location-based application, it also offers powerful development tool package for future use. By the help of Google Maps Javascript API, user can embed Google map into any web page or application. Currently, Version3 of Google Map Javascript API is in use, which provides faster response. To utilize service of Google Map, the first step is to connect with it. The development procedure is very simple. Web page can be written by HTML or HTML 5. At the beginning, Maps API Javascript is included by script tag. On the map layout, div element named by map_canvas is defined and location is determined which is used to hold Google Map. This map can be configured through defined Javascript object. Within this object, its properties can customize Google Map according to user's requirement. Then Google Map object is created and whole map is initiated when web page is loading.

4. RESULT

In this section, it discusses about the characteristic and setup of test bed. Description and explanation of match pattern used in this work is illustrated. Outline and procedure of web application is also included in this chapter and finally result is discussed.

4.1 Introduction to test bed

The wireless sensors are deployed in the different locations within Tampere University of Technology (Fig 26) [46]. The test bed classroom is located at the first floor of Konetalo building. Inner side classroom is selected as target room. Sensor network is distributed over classroom, office with different size and user utility, corridor and lab. Each room is approximately in the shape of square and inner side of building is a courtyard which offers more light into room. All the classrooms have ceiling and are without roof light as external lighting source.

Different parameters are measured by different type of sensors. Activity sensors and illuminance are only concerned in this work, so the distribution of it is shown in table. The activity sensors are either installed faced with door or point to busy area of user. Illuminance sensors are positioned to either face to window, which detects external lighting or face to lighting fixture to monitoring inner lighting condition. The height of location of activity sensor is 170 millimeter above ground so that it can cover larger area of activity zone. The height of illuminance sensors that measure external lighting is depend on the distance between window frame and ground floor. The height of illuminance sensors that monitor indoor condition is located around lighting source. In some room, more than one same type sensors are installed due to the division of the zone. The smaller subdivision of area, the higher accuracy of detective sensor is. Adaptor is installed within this room, and it collects environment data from wireless sensors and relay those information through Ethernet to database located in company.

Table 21. Sensor network distribution.

Room Number	Sensor Type	Sensor Node
Room1	Illuminance sensor	70
		305
		306
		308
		301
	Activity sensor	170
		161
Room2	Illuminance sensor	307
		142
		309
		302
		304
		303
	Activity sensor	132
		173
		142
Room3	Illuminance sensor	111
	Activity sensor	180
Room4	Illuminance sensor	97
	Activity sensor	134
Room5	Illuminance sensor	158
	Activity sensor	123
Room6	Illuminance sensor	120
	Activity sensor	120
Room7	Illuminance sensor	75
	Activity sensor	188

To test web application, computer room in TUT is selected as test bed. Layout of this room is displayed in the Figure 27. This room is accessed by student and used to provide computer usage for daily study. This room is opened whole day and user can freely enter and leave so that activity pattern in this room varies over time. Sensors area installed on the wall to monitor illuminance level of this room and activity sensor is used to detect activity occurred. Activity sensors are installed on inner side of this room so that interference from surrounding area is minimized. Illuminance sensor is installed on the level of working plane and provides accuracy measurement result.



Figure 27. Layout of computer room in TUT.

Wireless sensors used in this work are integrated sensors (Fig.28). There are illuminance sensor, activity sensor, humidity sensor, humidity sensor and carbon dioxide sensor integrated within one component. In this thesis work, only illuminance sensor and activity sensor are concerned. The address of sensor is set up by the control panel offered by device vendor. Accuracy of illuminance sensor is dependent on its position. To get correct result, position of illuminance sensor should be in the same level of working plane. To exclude environmental interference, obstacle between illuminance sensor and working plane is removed. As to activity sensor, high sensitivity can be both advantage and disadvantage. Highly sensitive activity can detect small movement occurred within this room and guarantee lighting system switched on immediately. The problem is that activity sensor may detect movement outside of detected area, which cause lighting to switch on, even there is no people inside. The position of activity sensor is selected deep inside classroom, which put distance between sensor and outside activity to minimize interference from unwanted source.



Figure 28. Integrate wireless sensor.

To implementation of En 15193-2007, characteristic of room is studied. Direct factors relate to energy usage are room and window dimension, location of windows, and power of lighting fixture. Those information are recorded in Table 22.

Table 22. Dimension information and basic information about test bed room.

Room dimension(m ²)	Width(m)		Length(m)	
74.3	8.2		9.06	
Window(m ²)	Width(m)	Height(m)	Distance(m)	Quantity
3.76	2.61	1.44	1.1	3
Total Power(W)	Power(W)		Quantity	
1044	58		12	

After obtain basic value of room characteristic, more comprehensive information is required. Information about lighting area is measured. Factors important to day light penetration are studied separately, including transparency index, depth index, linear factor, overhang factor, vertical factor, courtyard factor, façade factor and obstruction index. Those information are displayed in Table 23. This whole calculation process is

completed by the algorithm written in Java function. When controller of Spring framework receives input from user, it relay those information to functions. Complementary information is stored inside data base beforehand. Function also reaches out to data base to get extra information to complete calculation.

Table 23. Characteristic parameters of test bed room.

Zone depth(m)	Height of lintel(m)		Height of task area(m)		
3.6	1.54		1.1		
Daylight space(m ²)	Zone depth(m)		Zone width(m)		
9.4	3.6		2.61		
Transparency index	Façade area(m ²)		Work plane(m ²)		
0.4	11.3		28.2		
Depth index	Zone depth(m)		Window height(m)		
2.5	3.6		1.44		
Linear factor	Obstruction angle (°)				
0.8	24.5				
Overhang factor	Horizontal overhang angle(°)				
0.97	10				
vertical factor	Vertical fin angle (°)				
1	0				
Courtyard factor	Height of courtyard	Length of courtyard	Width of courtyard		
0.19	0.2	1.1	1		
Façade factor	Transmission factor	Frame factor	Dirt factor	Compensate factor	
0.85	1	0.7	1	0.85	
Obstruction index	Linear factor	Overhang factor	vertical factor	Courtyard factor	Façade factor
0.55	0.8	0.97	1	0.19	0.85

Finally, important numerical indicators are calculated from factor obtained above and referred to some complementary factors. Complementary factors include daylight factor classification, daylight factor, fenestration transmission, frame factor, dirt factor and compensate factor.

At first, intermediate factors are calculated, such as daylight factor classification, daylight factor, fenestration transmission factor, frame factor, dirt factor and compensate factor. At the end, daylight supply factor is derived. Result is shown in Table 24. Moreover, other characteristic factors are also automatically derived by web application.

Table 24. Important value of numerical indicators.

Daylight factor		Transparency index		Depth index		Obstruction index	
4.9		0.4		2.5		0.55	
Daylight factor classification	Daylight factor	Fenestration transmission	Frame factor	Dirt factor	Compensate factor		
2.4	4.9	0.82	0.7	1	0.85		
Daylight supply factor		a			b		
0.52295		1.2425			-0.0117		

4.2 Event processing of wireless sensor network

Occupancy sensors have been utilized as control devices in different applications of building automation. It has huge potential in reduction of energy waste by switching off the electric load when users leave their occupied area. Jennings and others found that in private offices occupancy sensors that turning the lights off after a 15-20 minute period of no-occupancy saves between 20-26% in lighting energy compared to the manual operation of a wall switch alone [47].

The average lighting energy saving strongly depends on type of space and time delay. Study of types of space can obtain the occupancy pattern, and corresponding control schema can be designed. Time delay means time that light will remain on after the last motion is detected. The small delay time can lead to reduction of energy consumption. However, if delay time is too small, it will cause abnormal switch off even when user is still staying within the occupancy area which can be annoying. On the other hand, if value of delay time is large, effect of energy saving will decrease, since light remains on even when user leaves. So occupancy sensors which can be programmed with different delay time in different occasions have been applied in many applications. But in this thesis work, study of activity pattern is beyond the scope, so that delay time of occupancy sensor may not be programmed according to demand. The delay time of this application is set to predefined value.

After connection to company database, data is transferred in the format of XML. This information includes all sensor information from the university sensor network. The extra useless message is filtered by the Esper expression.

To test connection between application and database of company and configuration of Esper, preliminary query for activity sensor and illuminance sensor is implemented.

As to activity sensor, node id, sensor type, time stamp, activity percent, value of counter, value of activity, and classroom number are required. To obtain that information, the Esper query clause is written as Appendix A.

The received message is in format of XML, which is type of structure language. To extract desired information from whole section, step-by-step query strategy is adapted. The root element is Network and all descendants of root element can be accessed with proper navigation. In addition, this query expression is very flexible, so different parameters or value important to the research work can be obtained with little

modification. The sequence of parameter can not be changed arbitrarily, since it is confined by the listener.

As to illuminance sensor, node id, sensor type, time stamp, illuminance value, and classroom number are required. The Esper search expression can be written in the format as Appendix B.

Due to the result of preliminary query, all of information from all sensors is gathered. But not all the activity sensor is interest to the project purpose, since the preliminary data does not have logic connection. In this step, the raw data is grouped into relevant data by sensor type and node id. Comparison between instantaneous value and threshold is also inspected. The query clause for identifying appropriate sensor consists of two events. The first event is to keep collecting all the activity sensors. The second event is to identify the same type of sensor of same node id with change value greater or less than the threshold value. c

The Esper engine will keep inspecting event pattern which meets the rule. The match pattern used in this clause can ruled out abnormal situation that is to guarantee that only two successive sensor data is compared, so that the possibility of accumulated change greater than threshold is eliminated.

As to the illuminance sensor, the instantaneous value change in illuminance is required to be detected. Not as the activity sensor, there are numbers of illuminance sensors are deployed within one room, which causes the match pattern confuse and generates wrong result. This problem is solved by utilizing and-not clause. Information collection of illuminance sensor with instantaneous value greater than threshold is shown in Appendix D. The mechanism is described as following. This query clause will define three events. The first event is inspecting all illuminance sensors on the network, and the second event is to search for illuminance sensor with instantaneous positive or negative change value greater than threshold. The third event is to record the illuminance sensor whose node id is same as the illuminance sensor from the first event. As long as the first event is captured by engine, it will store in the memory and try to find the second event to make pair. If the second event is detected and the third event does not occurs, all of first events are related to the second event and the match pattern will restart in the next round. If the first event and third event happen before second event, this clause is stopped and restarts all over for new pattern search.

Code displayed in Appendix E is used to define the delay time of the occupancy sensor. Due to the limited capability of sensor, it cannot be pre-programmed according to the activity study or self-adaption to surrounding environment. The delay time value is set to 20 minutes according to the best practice from many researches. All the sensors are group by their id, so the status of all the target sensors is monitored. Once the message flow matches the pattern that is the initial value change of activity sensor greater than zero in the next twenty minutes, and there is no positive value change from activity sensor that means termination condition is satisfied, the result will be forward to control schema clause to stop it. If the time interval is no longer than 20 minutes, this query pattern is restarted.

The manual control schema clause is used to capture message that satisfies the requirement. Coding is shown in Appendix F. Once the positive change of activity

sensor occurs, the schema will keep polling the illuminance sensor within the same room that if positive change also happens. Due to the group-by clause, every activity and illuminance sensors meeting requirement are inspected. Within one room, more than one activity sensor or illuminance sensor may exist, the phenomena of message mixture may lead to the wrong result. But this situation is avoided since the pair match is completed in the previous query clauses. If those two conditions are satisfied, it starts to receive the energy impulse and accumulates the energy consumption. Once the illuminance sensor within the same room has negative value, search pattern is quitted and restart for next search period.

As to the automatic control schema, start condition is activity sensor with instantaneous positive change. The search engine keeps looking for next value of activity sensor and result from terminate clause. The accumulated addition continues until there is no new activity sensor with positive value or the terminate clause turns into true. Coding of automatic control is shown in Appendix G.

As to manual-on-automatic-off schema, this query pattern keeps looking for positive value as start condition. Once the positive value of illuminance sensor within the same room also happens, the accumulated addition starts. Once the activity sensor turns into negative value, the addition stops. Correspondent coding is shown in Appendix H.

4.3 Implementation of Web application

Web application is built on Spring MVC framework. When user starts application, user is first directed to log in web page. Google map is automatically opened up which displays current location of user and coordinate of geolocation is stored into database for further calculation.

This information is essential for energy calculation, since daylight duration is directly related to the geolocation of that country. Latitude of Finland is 60 degree. The characteristic of this country is to have long day lighting duration during summer and very short daylight time in winter. The uneven lighting condition throughout the year means that lighting consumption monitoring is more important for Finland. During summer, external lighting supply is ample, so that demand for usage of lighting decreases. In this case, users leave room without switching off the light will cause more energy waste than winter. In the winter, energy consumption from air condition, ventilation will increase in comparison with summer. To lower burden of electricity supply during winter, energy saving from lighting consumption is good solution to this problem by monitor and control lighting usage. This interface is shown in Figure 29. Location of TUT is shown at the central of the map by red tag.



Figure 29. Local map and geography coordinates.

Log in form is displayed under log in interface which is used to accept user name and password. If username and password is verified against data in the database, user is redirected to parameters input interface. Layout of interface is displayed in Figure 30.

click here'."/>

Figure 30. Log in interface.

On user input page, different numerical indicators are grouped into separate table by parameters needed being provided by user, which are demonstrated in Figure 31. When user provides all necessary information, web application will provide standard energy

consumption by EN15193. Result is displayed at the bottom of this page. This form is reusable; user can enter information of multiple rooms on this page. On the left bottom, recommend energy consumption is displayed. Room number, energy consumption of lighting, energy consumption of parasitic energy, total energy consumption and energy indicator is listed within this table. On the right bottom, total lighting usage, total parasitic usage, total energy and numerical indicator calculate from wireless sensor network is displayed.

Information about energy consumption of lighting, energy consumption of parasitic energy, total energy consumption and energy indicator is dynamical displayed. This updated period is configured by Ajax call function, so that user will observe change of energy consumption within specific room.

After users choose target room, they can still modify their choice, by either switching room number or adding more room

Theoretically, energy consumption of room depends on dimension characteristic, outdoor lighting condition and activity pattern. Smaller size and seldom visited room with ample exterior lighting consume less energy.

Different set of indicators are grouped under different form. Room information describes configuration of test bed room. Transparency index indicates area of room benefited from outside lighting source. Daylight factor show the influence of room from lighting penetration. In the form of daylight dependent artificial lighting control, control schema is determined. There are three type of control schemas are available for user to choose, that are manual control schema, automatic control schema, and automatic-manual control schema. Control schema from daylight dependent artificial lighting control determines level of function of daylight penetration. From form of occupancy dependency lighting control system factor, value of control schema is used to assign value of occupancy dependency factor. In the operating hour form, building type decides theoretical value of operating hour. In the form of luminaire maintenance factor and parasitic energy, maintenance factor and energy is calculated. Factors required to calculate daylight zone are specific in the form of depth of zone, daylight zone and depth index. Obstruction factors are derived according to the input from form of obstruction index. Daylight supply factor is dependent on maintenance illuminance value which is collected from user input in the form of daylight supply factor. Monthly daylight dependent factor is conversion by the information of form of daylight dependent factor. By specifying value in the form of absence factor, absence indicator is derived. To obtain lamp lumen maintenance factor and lamp survival factor, lighting source and hour of use is filled out in the form. Room index is deduced from value of form of room surface maintenance factor.

Room Informaiton				
Luminaire Power:		Quality:		
Classroom:				
Transparency Index				
Area of Façade Opening:		Work Plane:		
Daylight Factor				
Windo Type:				
Daylight Dependent Artificial Lighting Control				
Control Schema:				
Occupancy Dependence Lighting Control System Factor				
Control Schema:				
Operating Hour				
Building Type:				
Luminaire Maintenance Factor and Parasitic Energy				
Luminaire Type:		Clean Interval:		
Parasitic Power:		Parasitic Number:		
Charging Power:		Charging Number:		
Room	LightingEnergy	Parasitic Energy	Energy Consumption	Energy Indicator
Depth of Zone, Daylight, Depth Index				
Height of Lintel:		Height of Task Area:		
Depth of Daylight Zone		Width of Daylight Zone:		
Obstruction Index				
Obstruction Angle:		Overhang Angle:		
Fin Angle:		Height Courtyard:		
Length of Courtyard:		Width of Courtyard:		
Daylight Supply Factor				
Maintenance Illuminance:				
Daylight Dependent Factor				
Monthly distribution:				
Absence Factor				
Building Type:		Room Type:		
Lamp Lumen Maintenance Factor, Lamp Survival Factor:				
Lighting Source:		Hour of Use:		
Room Surface Maintenance Factor				
Length of Room:		Width of Room:		
Mounting Height:		Luminaire Strategy:		
Clean Interval:				
Total Lighting Energy	Total Parasitic Energy	Total Energy	Total Indicator	

Figure 31. User interface for parameters input.

When user is directed to visualization page, energy consumption of different control schema is compared. There are two different figures shown different information. On the top of webpage, historic energy consumptions of different control schemas are demonstrated. This can be seen in the Figure 32. Due to the limit of device, lighting within one room is presumed to switch on and off at the same time. Initially data from different sources are transferred in the format of XML. That information are collected

and converted to Java Object by Esper listener. Further, objects are mapped into rows of database. Web application will connect to database to fetch back of data, when user send request. This time reverse conversion process occurs. Different schemas are represented by different colour lines. Manual control schema is displayed with blue line, auto control schema is with red line, manual-on-automatic-off schema is indicated by yellow line and standard is represented by green line. In this figure, data is fetch from data base, in which historical data is stored. Information from this figure reveals activity pattern of this particular room in the past, which can be used to predict future pattern. By investigating of activity pattern, more accuracy control can be derived. From comparison of different colour of lines, the most optimal control schema is obtained as well. Green line standard for benchmarking and energy usages from different control schemas are compared against it. Value of benchmarking is average value of monthly calculated recommended value. Same room on different days has different energy consumption. We can see that benchmarking of average energy usage for this room is 5282 Wh. But there are different energy performances among different schemas. Manual control schema is the most deviation from benchmarking. Its maximum value goes above 12000Wh and minimum value is larger than benchmarking value. It can be deduced that users may not turn off the lighting, when they leave the room. The prolong lighting duration causes extra energy consumption. Energy consumption based on automatic and manual automatic schema follows the same trend. Sometimes consumption is above benchmark value, which can go close to 8000 Wh. Sometimes usage can be lower than 4000. This fluctuation around benchmarking can be contributed to the calculation of benchmark value, which is the average of energy consumption throughout the whole month. After comparison, it can be concluded that Manual-automatic control schema is the most suitable way to control lighting in this room according to activity pattern.

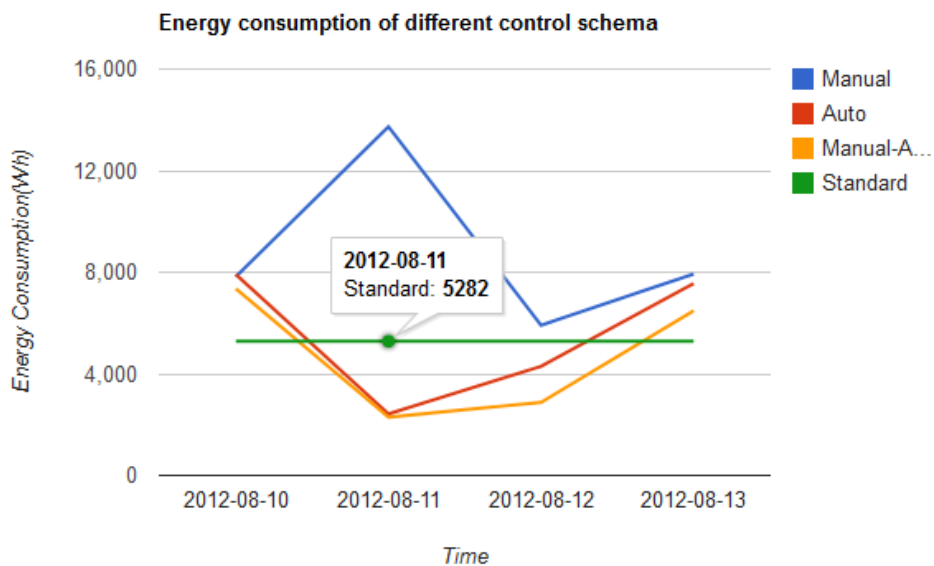


Figure 32. Historic energy consumption of different control schemas.

In figure 33, monitored energy consumption of different control schemas by hour is shown by column chart. Blue colour column represents manual control schema, red colour column indicates automatic control schema, and yellow one shows manual-automatic control schema. Data in this figure are kept continuous updated. On the client side, Ajax schema will keep sending request to server side to get latest data in the database. On the server side, it receives the request, connect to data base and generate response embedded target data in the form of Json. This data is obtained by Ajax object and transfer this value to data table of Google chart, which will update chart to display latest result. Each chart is assigned to each room. If user enters more than one specification in the parameter input interface, multiple of charts are displayed correspondently. The scale of energy consumption is automatically implemented. From figure 33, it is clear that different control schema consumption different level of energy. In addition, energy consumptions of different control schema vary with time. From 17:00-18:00, energy usage for that room is reached to its maximum consumption. This can be explained by the fact that supply of sunlight from outside is decreased due to night time coming. It is also true that people have more activity for that room from this time interval. Then energy usage stats declining, which begin from 19:00 o' clock. This phenomenon can be explained that students are finishing their study and leaving university. During this period time of monitoring, energy usage reaches to its minimum at around 21:00 o' clock. Even night time is coming; most of students are already leave from university. There is only intermittent activities occurred within detect area. There is no universal solution for selection of control schema. It varies from case to case. By comparison of them, the most optimal control schema can be selected for that particular room. From Figure, column with yellow colour has the lowest energy usage. In this case, manual-auto control schema is the one that should be implemented for this room.

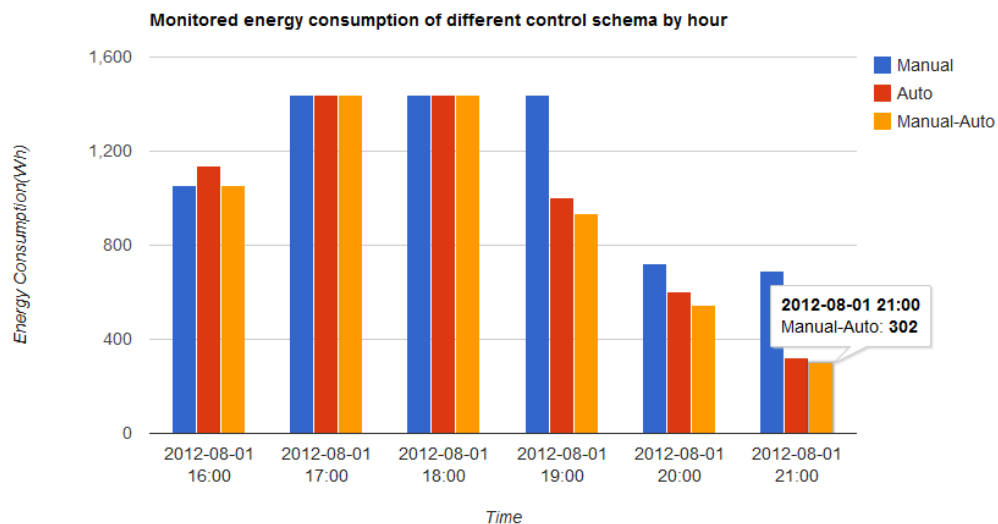


Figure 33. Energy consumption of different control schemas by hour.

5. CONCLUSIONS

This chapter presents brief review on the objective of work, implementation of system and evaluation of the result. In this section, future opportunity within this field is also discussed.

5.1 Conclusion on the thesis work

Objective of this thesis work is to develop web application which can be utilized to monitor energy consumption of commercial building and derive optimal control schema for the building. This application will provide benchmark value of specific room according to energy requirements for lighting drafted by EU. Different energy consumptions by different control schemas can be monitored by wireless sensor network. Information aggregation, analysis and display is implemented by web application.

To reach this objective, design of web application is selected to realize the design. There are several steps needed to be completed, including configuration of wireless sensor network, information communication between host computer and database in the company, information aggregation and analysis, database management, and implementation of web application. Wireless sensor work for monitoring activity pattern is installed in the different location of TUT. Activity occurred within room is collected and send to adapter by wireless sensor network and is forward to database of company by Ethernet. Host computer builds up communication tunnel to company by TCP/IP protocol and data is successfully transferred to host side. Discrete information is filtered first and relevant data is grouped and matched according to predefined pattern. According to schema definition, different action is triggered and data are stored into database. User activates this application by offering correct user name, password and required input parameters. After that, reference energy usage is provided according to EN15193. On display page, Google chart visualizes result from wireless sensor network in the form of bar chart and line chart. On that page, different energy usage from different control schema is displayed on hourly basis and daily basis. Optimal solution is derived from comparison.

Integration of wireless sensor network with web application provides new solution to monitoring energy consumption of commercial building. First, new system demonstrates excellent compatibility over traditional building automation device. Information transferred between wireless sensor network and web application is in the unified format of XML. The same semantic of message enable seamless information sharing among different parts of system. Secondly, new system is user friendly oriented. New wireless sensor network can be easily set up by configuring some basic network

information, such as node id, address. This can directly speed up the configuration of new system. New sensor can be easily integrated into existing configured by control panel as well. Web application can be conveniently expanded to accommodate more information from new sources correspondently. Thirdly, new system offers great scalability. Different wireless sensors are connected by wireless network. Capability of individual network can be expanded by configuring new sensor node. Whole system can scale up by integration of different individual network to form new one. As to web application, with little modification can be adjusted to new wireless network. Fourthly, system provides good portability. Sensor can be easily reinstalled to different position according to measurement requirement, since wireless communication eliminates hard wire work. Web application can enable any user access to monitoring interface, as long as user can connect to internet. In this way, platform dependent control panel is replaced. Fifthly, whole system is easy to be maintained. There is no extra requirement on client side, when update of application occurs. Update process took place on the server side can be synchronized on client side. This saves trouble to update every client terminal manually. Sixthly, system display good dynamic behaviour. New sensor can plug in and out to form dynamic behaviour of sensor network. This is especially true for web application, which built on MVC framework. Application can be easily expanded and trimmed by adjustment of controller, view and model.

This thesis work navigates through every step of design and implementation of new monitoring system from scratch. In the initial stage, problem domain is studied and BS EN 15193 is proposed to determine energy requiremnt of commerical building. In the next step, extracting actual energy consumption information from field devices is implmented by wireless sensor network and communication. In the last stage, comparision between benchmark value and actual energy situation is achieved by utilizing web appliatoin. The implmented system provides detail information about energy consumption of commercial building by hourly and daily based. Energy performances of different schemas are recored in the chart, which can be used as reference to design optmail control schema. Historical data can be also fetched to study activity pattern and user behavior in the past. The actual energy usage is accurately simulated by the web applicatoin and reflect the activity pattern.

5.2 Future work

The utilization of the wireless sensor network can avoid unnecessary energy waste. To optimize the performance of the system, activity pattern should be studied and occupancy sensor with high quality can be utilized. The time schedule derived from activity pattern can be used as reference to program the occupancy sensor or the occupancy sensor can automatically adjust to the specific occasion. In this case, delay time will be in the optimal range and extra energy consumption can be reduced to the minimum.

Match-recognize and pattern-matching expression of Esper are applied in this work to extract and correlate useful message from multiple information sources. The accuracy of calculation of real time energy consumption may be increase by utilizing other

commercial calculation model.

Higher demand has been imposed on the lighting system, since energy usage is not the only factor to be taken consideration. The lighting quality, such as vision performance, subjective behaviour, health and aesthetical effect has been highly valued. This study has shown that productivity can be related to the quality of illumination. Although higher requirement of lighting supply will consume extra power, overall economic benefit gets improved.

The energy consumption of commercial building mainly consists of lighting, heating, ventilation and air conditioning. In this thesis work, main focus is monitoring the lighting consumption. To improve energy efficiency of entire commercial building, other three factors will be required considering. The comprehensive study of energy usage of commercial building by wireless sensor network can be very promising area of research in the future.

References

- [1]Takuya YOKOMIZO , Ichiro TAKANO, Miki KOBAYASHI, "Effective Energy Saving on the Campus Classroom Lighting by Scheduling Management" in Proc of TENCON 2009 - 2009 IEEE Region 10 Conference, 23-26 Jan. 2009, Singapore, pp1-6.
- [2]Mark S. Rea, Ph.D, "The IESNA Lighting Handbook Reference and Application, 9th Edition". NewYork: IESNA, 2000.
- [3]Joan L. Pellegrino , A. Hunter Fanney , Steven T. Bushby, "Measurement science roadmap for net-zero energy buildings workshop summary report", NIST Building and Fire Research Laboratory, March 2010.
- [4]Wouters, "The implementation of energy efficient buildings policies in 5 continents", opportunities offered by the BUILD UP interactive web portal. Brussels, Belgium, 14 October 2009.
- [5]M. Santamouris, E. Dascalaki, "Passive retrofitting of office buildings to improve their energy performance and indoor environment: the OFFICE project, Building and Environment", 37 (2002) 575–578.
- [6]A. Lagoudi, M. Loizidou, M. Santamouris, D. Asimakopoulos, "Symptoms experienced, environmental factors and energy consumption in office buildings, Energy and Buildings", 24 (1996) 237–243.
- [7]European Council, "Directive 2002/91/EC of the European parliament and of the council on the energy performance of buildings", Official Journal of the European Communities 2002.
- [8]B.Anderson, "CEN/BT WG 173 EPBD Version V5-Explanation of the general relationship between various CEN standards and the Energy Performance of Building Directive (EPBD) (“Umbrella document”)", December 2005.
- [9] BS EN 12193:2007, "Light and Lighting", 25 October,2007, Policy and Strategy Committee
- [10] EN 12464-1, “The Lighting of Workplaces”, Excellent Lighting,Saving Energy, 2002

- [11] EN 1838, "Lighting applications – Emergency lighting" , Excellent Lighting,Saving Energy, 1999
- [12] EN 60598,"Luminaires-general requirements and tests", European Committee for Electrotechnical Standardization, 1, April, 2006
- [13] EN 60570, "Electrical supply track systems for luminaires", European Committee for Electrotechnical Standardization, 24, July, 2003
- [14] EN 61347, "Lamp controlgear - General and safety requirements", European Committee for Electrotechnical Standardization, 2008
- [15] BS EN 15193, "Energy performance of buildings — Energy requirements for lighting", 30 November 2007, England, Policy and Strategy Committee.
- [16] Wilfried Pohl, Matthias Werner, "Comfort monitoring for CEN standard EN15251 linked to EPBD (COMMONCENSE)", May 2010
- [17] Peter Boyce, Peter Raynham, "The SLL Lighting Handbook" , Chartered Institution of Building Services Engineers; 1st edition (23 April 2009), pp 314.
- [18] Kristina Jucyte , Karolis Kevelaitis, Sung Won Park, "Web service implementation with SOAP and REST", Roskilde University, Jan 2006.
- [19] Roy Thomas Fielding, "Architectural Styles and the Design of Network-based Software Architectures", Architectural styles and the design of network-based software architectures, University of California, Irvine 2000
- [20] REST vs SOAP Web Service, October 11, 2009, available online:
<http://nucleus-inc.blogspot.fi/2009/10/rest-vs-soap-web-services.html>
- [21] Luigi Martirano IEEE member, "Lighting systems to save energy in educational classrooms", Environment and Electrical Engineering (EEEIC), 2011 10th International Conference, pp 1-5, Rome Italy, 8-11 May 2011
- [22] Luigi Martirano, "A Smart Lighting Control to Save Energy", Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), 2011 IEEE 6th International Conference, pp, 132 – 138, Prague, Czech Republic, 15-17 Sept. 2011
- [23] R. Faranda, S. Guzzetti and S. Leva, "Design and Technology for Efficient Lighting", Paths to Sustainable Energy, Jatin Nathwani and Artie Ng (Ed.), 2010.
- [24] Roman Szczepaniak, Michael Wilson, " Investigating Energy Requirements for Lighting: A Critical Approach to EN15193", Department of Architecture and Spatial Design of London Metropolitan University, 9 Apr 2010

- [25] Moghavvemi M, Seng LC, "Pyroelectric infra-red sensor for the intruder of detection" ,TENCON 2004. 2004 IEEE Region 10 Conference , 656 - 659 Vol. 4, 21-24 Nov. 2004
- [26] Benya J, Heschong L, McGowan T, Miller N, Rubinstein F. Advanced Lighting Guidelines. White Salmon (WA): New Buildings Institute, 2003.
- [27] I.F. Akyildiz, Weilian Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," Communications Magazine, IEEE, vol. 40, no. 8, pp. 102–114, Aug 2002.
- [28] Teemu Laukkarinen, Jukka Suhonen, Timo D. Hämäläinen, Marko Hännikäinen, "PILOT STUDIES OF WIRELESS SENSOR NETWORKS: PRACTICAL EXPERIENCES", Design and Architectures for Signal and Image Processing (DASIP), 2011 Conference, pp 1-8, Tampere, Finland, 2-4 Nov. 2011.
- [29] Wirepass Ltd, "Wirepas Ltd. Sensor Data Feed and Actuator Interfaces Specification v. 1.1 5", Tampere, Finland, 12.2011.
- [30] Luckham, David C, "Event Processing for Business: Organizing the Real-Time Enterprise", Hoboken, New Jersey: John Wiley & Sons, Inc., p. 3. ,2012
- [31] Event Stream Processing - A New Physics of Software, available online:
<http://www.information-management.com/infodirect/20050729/1033537-1.html?zkPrintable=1&nopagination=1>
- [32] Principle of Esper, available online:
<http://esper.codehaus.org/tutorials/tutorial/tutorial.html>
- [33] JDBC overview, available online:
<http://www.oracle.com/technetwork/java/overview-141217.html>
- [34] Structure of model, view and controller available online:
<http://java.sun.com/blueprints/patterns/MVC-detailed.htm>
- [35] Struts 1 framework MVC, available online:
<http://www.raistudies.com/struts-1/architecture-of-struts-1-mvc-framework/>
- [36] Struts 2 MVC architecture, available online:
<http://www.roseindia.net/struts/struts2/struts-2-architecture.shtml>
- [37] Gary Mak, Ken Sipe, Josh Long, "Spring Recipes - A Problem-Solution Approach - 2nd Edition", Apress, pp1104, Aug 2010
- [38] Description about web service description language, available online:
<http://www.w3.org/TR/wsdl>

- [39] Definition of web service description language, available online:
http://en.wikipedia.org/wiki/Web_Services_Description_Language
- [40] Description about Java Architecture for XML Binding, available online:
<http://www.oracle.com/technetwork/articles/javase/index-140168.html>
- [41] Basic principle of IOC, available online:
<http://static.springsource.org/spring/docs/2.5.3/reference/beans.html>
- [42] Processing workflow in Spring Web Service, available online:
<http://static.springsource.org/spring-ws/site/reference/html/server.html>
- [43] Optimizing web application with AJAX available online;
<http://www.codeproject.com/Articles/16272/Optimizing-Web-Applications-with-AJAX-and-Spread>
- [44] Description of Tile framework, available online:
<http://tiles.apache.org/framework/tutorial/pattern.html>
- [45] Introduction to use google chart, available online:
<https://google-developers.appspot.com/chart/interactive/docs/index>
- [46] Building layout of Konetalo of University of Tampere, available online:
https://www.tut.fi/idcprod/groups/sg_authenticated/@1102/@web/documents/doku/p017094.pdf
- [47] Galasiu, A.D.; Newsham, G.R.; Suvagau, C. Sander, D.M, "Energy saving lighting control systems for open-plan offices: a field study", v. 4, no. 1, pp. 7-29, July 2007

A. APPENDIX A – Script for Activity Sensors.

```
String activity sensor = "select " +  
"Network.Node.Sensor.Measurement.Component[0] as percent,  
Network.Node.Sensor.Measurement.Component[1] as counter,  
" +  
"Network.Node.Sensor.Measurement.Component[2] as value,  
Network.Node.name as room," +  
"Network.Node.Sensor.Measurement.time as time, " +  
"Network.Node.id as id, " +  
"Network.Node.Sensor.Measurement.quantity as sensor_type  
from SIDF" +  
"where Network.Node.Sensor.Measurement.quantity=\"Activi  
ty monitor\" ) ";
```

B. APPENDIX B – Script for Illuminance Sensors.

```
String illuminance sensor = "select" +  
"Network.Node.Sensor. Measurement.Component as value, " +  
"Network.Node.name as room," +  
"Network.Node.Sensor. Measurement.time as time," +  
"Network.Node.id as id, " +  
"Network.Node.Sensor.Measurement.quantity as sensor_type  
from S IDF" +  
"where Network.Node.Sensor.Measurement.quantity=\" Illumina  
nce\" ) ";
```

C. APPENDIX C – Script for Value of Activity Sensor.

```
String stmt_act = "insert into MOTION
select A.network[0].node[0].name as act_room," +
  "A.network[0].node[0].sensor[0].measurement[0].quantity as
  sensor_type_act, " +
  " A.network[0].node[0].sensor[0].measurement[0].time as act
  _time, " +
  " lighting.esper.IncConverter.caIntConverter( A.network[0].n
  ode[0].sensor[0].measurement[0].component[0].value, B.netwo
  rk[0].node[0].sensor[0].measurement[0].component[0].value)
  as act_start, B.network[0].node[0].id as      act_id " +
  " from pattern" +
  " [every A = SI(network[0].node[0].sensor[0].measurement[0].
  quantity=\"Activity monitor\")->B= SI(network[0].node[0].se
  nsor[0].measurement[0].quantity=\"Activity monitor\" +
  " and ( lighting.esper.IncConverter.caIntConverter(A.network
  [0].node[0].sensor[0].measurement[0].component[0].value, ne
  twork[0].node[0].sensor[0].measurement[0].component[0].valu
  e)>10 " +
  "or lighting.esper.DecConverter.decConverter(A.network[0].n
  ode[0].sensor[0].measurement[0].component[0].value, network
  [0].node[0].sensor[0].measurement[0].component[0].value)>10)
  " +
  " and A.network[0].node[0].id=network[0].node[0].id) " +
  " and NOT SI(network[0].node[0].sensor[0].measurement[0].qu
  antity=\"Activity monitor\")==\" +
  " and A.network[0].node[0].id=network[0].node[0].id)]" +
  "group by A.network[0].node[0].id";
```

D. APPENDIX D – Script for Value of Illuminance Sensor.

```
String stmt_ill = "insert into LIGHTING
select lighting.esper.IncConverter.caIntConverter
(A.network[0].node[0].sensor[0].measurement[0].component[0].
value, " +
"B.network[0].node[0].sensor[0].measurement[0].component[0].
value) as ill_start, " +
" A.network[0].node[0].name as ill_room, " +
"A.network[0].node[0].sensor[0].measurement[0].time as ill_
time, " +
"A.network[0].node[0].sensor[0].measurement[0].quantity as
sensor_type_ill, " +
"B.network[0].node[0].id as ill_id " +
" from pattern " +
"[every A = SI(network[0].node[0].sensor[0].measurement[0].
quantity=\"Illuminance\")->B = SI(network[0].node[0].sensor
[0].measurement[0].quantity=\"Illuminance\" " +
" and (lighting.esper.IncConverter.caIntConverter(A.network
[0].node[0].sensor[0].measurement[0].component[0].value, ne
twork[0].node[0].sensor[0].measurement[0].component[0].valu
e)>1 or lighting.esper.DecConverter.decConverter(A.network
[0].node[0].sensor[0].measurement[0].component[0].value, ne
twork[0].node[0].sensor[0].measurement[0].component[0].valu
e)>1) " +
"and A.network[0].node[0].id=network[0].node[0].id) " +
"and NOT SI(network[0].node[0].sensor[0].measurement[0].qua
ntity=\"Illuminance\" " +
"and A.network[0].node[0].id=network[0].node[0].id)]" +
" group by A.network[0].node[0].id";
```

E. APPENDIX E – Script for Configuration of Delay Time.

```
String terminate = "insert into TERMINATE select A.act_id as  
act_id from pattern [every A = MOTION(act_start<0)-> (timer:  
interval(1200 sec) and not MOTION(act_start>0))] group by  
A.act_id";
```

F. APPENDIX F – Script for Consumption by Manual Control.

```
String manual = "select A.act_room as room,A.sensor_type_act  
  as sensor_type, A.act_time as time,A.act_start as value, A.  
act_id as id, C.power as power" +  
" from pattern" +  
"[every A=MOTION(act_start>0)-> B=LIGHTING(A.act_room = ill  
_room and ill_start>0) ->every C=ENERGY and not LIGHTING(ill  
_start<0 , B.ill_room =ill_room)] group by A.act_room" ;
```

G. APPENDIX G – Script for Consumption by Automatic Control.

```
String auto = "select A.act_room as room,A.sensor_type_act a  
s sensor_type, A.act_time as time,A.act_start as value, A.ac  
t_id as id, C.power as power" +  
" from pattern" +  
"[every A=MOTION(act_start>0)->(every C=ENERGY and not MOTI  
ON(act_start>0, A.act_id= act_id)) and not TERMINATE(A.act_  
id= act_id)] group by A.act_id ";
```

H. APPENDIX H – Script for Consumption by Manual-automatic Control.

```
String manualOn ="select A.act_room as room,A.sensor_type_a  
ct as sensor_type, A.act_time as time,A.act_start as value,  
A.act_id as id, C.power as power" +  
" from pattern" +  
"[every A=MOTION(act_start>0) -> B=LIGHTING(A.act_room = il  
l_room and ill_start>0) -> every C=ENERGY and not MOTION(act  
_start<0 , A.act_room =act_room)]";
```