

WARSAW UNIVERSITY OF TECHNOLOGY
FACULTY OF ELECTRICAL ENGINEERING
INSTITUTE of CONTROL and INDUSTRIAL ELECTRONICS



MS THESIS
Course: Computer Science

Mateusz PIWEK
Student's book no.: 181 581

Academic year: 2010/2011
Warsaw, 30.11.2010

Title:

WEB BASED USER'S APPLICATIONS FOR NA61/SHINE EXPERIMENT AT CERN

The scope of the thesis:

1. Introduction
2. CERN Technology and Knowledge Transfer
3. The NA61/SHINE experiment
4. Work environment
 - Apache Cocoon project
 - Eclipse development environment
 - PostgreSQL project
 - Elog application
5. Independently developed Web-based applications
 - ShiftsScheduler
 - ElogLooker
6. Conclusions

Supervisor:

Head of the Division

(doc. dr inż. Krzysztof Amborski)

(dr hab. inż. Bartłomiej Beliczyński, prof. PW)

The thesis is due to be submitted by: 15.09.2011

Accomplished and defended thesis
remains the property of the Institute, Division
and will not be returned to its author.

POLITECHNIKA WARSZAWSKA
WYDZIAŁ ELEKTRYCZNY
INSTYTUT STEROWANIA I ELEKTRONIKI PRZEMYSŁOWEJ



PRACA MAGISTERSKA
na kierunku: Informatyka

Mateusz PIWEK
Nr imm.: 181 581

Rok. akad.: 2010/2011
Warszawa, dn. 30.11.2010 r.

Temat:

APLIKACJE UŻYTKOWE OPARTE NA WWW DLA EKSPERYMENTU NA61/SHINE W CERNie

Zakres pracy:

1. Wprowadzenie
2. CERN a transfer wiedzy i technologii
3. Eksperyment NA61/SHINE
4. Środowisko pracy
 - Projekt Apache Cocoon
 - Środowisko programistyczne Eclipse
 - Projekt PostgreSQL
 - Aplikacja Elog
5. Samodzielne aplikacje webowe
 - ShiftsScheduler
 - ElogLooker
7. Wnioski

Kierujący pracą:

Kierownik Zakładu Sterowania

(doc. dr inż. Krzysztof Amborski)

(dr hab. inż. Bartłomiej Beliczyński, prof. PW)

Termin wykonania: 15.09.2011

Praca wykonana i zaliczona pozostaje własnością Instytutu
i nie będzie zwrócona wykonawcy

Table of Contents

1	Introduction.....	1
1.1	About European Organization for Nuclear Research – CERN.....	1
1.1.1	Information technologies at CERN.....	1
1.2	CERN Technology Transfer.....	2
1.3	NA61/SHINE experiment.....	3
1.3.1	NA61/SHINE Science.....	3
1.3.2	Work in NA61/SHINE.....	3
1.4	My contribution to NA61/SHINE.....	4
1.4.1	Software requirements.....	4
1.4.2	NA61/SHINE work organisation with ShiftsScheduler.....	4
1.4.3	Run statistics with ElogLooker.....	4
1.4.3.1	Elog and ElogLooker.....	4
2	Requirements for applications.....	7
2.1	ShiftsScheduler.....	7
2.1.1	Shifts.....	7
2.1.1.1	Shifts reservations and confirmations.....	8
2.1.2	Account types.....	8
2.1.3	Reservations on behalf of another person's.....	8
2.1.4	Reservation cancelling vs. rejecting.....	10
2.1.5	Automatic confirmations.....	10
2.1.6	Blocs.....	11
2.1.6.1	NA61 blocks layout.....	11
2.1.7	Points awarding.....	13
2.1.7.1	Points awarding rules.....	13
2.1.7.2	Points awarding examples.....	15
2.2	ElogLooker.....	18
3	Used technologies.....	19
3.1	WWW – presentation tier.....	19
3.1.1	URL.....	20
3.1.2	HTML.....	21
3.1.2.1	CSS.....	23
3.1.2.2	ECMAScript.....	24
3.1.2.3	DOM model.....	24
3.1.2.4	XML.....	25
3.1.2.5	XSLT.....	27
3.1.2.6	AJAX.....	28
3.1.3	HTTP.....	29
3.2	Java – application tier.....	29
3.2.1	Apache Cocoon Framework.....	30
3.3	Databases – storage tier.....	31
3.3.1	PostgreSQL.....	31
3.4	Eclipse – development environment.....	32
4	User's interface.....	33
4.1	Shifts Scheduler.....	33
4.1.1	Run option.....	33
4.1.2	Statistics.....	35
4.1.3	Contributors.....	36
4.1.4	Personal informations.....	36
4.2	Elog Looker.....	37
5	Conclusions.....	39

1 Introduction

The Master Thesis titled „Web based users applications for NA61/SHINE experiment at CERN” presents the World Wide Web technologies that has been used during development of the software suite to support work organisation and data interpretation in NA61/SHINE experiment. Presented software was implemented and is use by a group of approximately sixty users¹.

The NA61/SHINE is one of many projects that takes place in the European Organization for Nuclear Research (CERN) located near Geneva.

1.1 About European Organization for Nuclear Research – CERN

CERN (French: Organisation européenne pour la recherche nucléaire²) was established on 29th of September 1954. Poland is a member state since 1991, however for a long time before joining CERN Poland as the only country of Communist Block had an observatory status. Nowadays Polish scientists are taking a part in a main CERN's experiments such as ALICE, ATLAS or CMS.

CERN's essential scientific facilities are the particle physics accelerators and detectors. The beam provided by the accelerator or collider by interacting with a target or another particle beam triggers the production of a new particles. These particles – so called products of collisions are registered by detectors, the detecting system consists of a set of different purposes detectors, such a system provides a very valuable scientific data.

The largest and the most powerful CERN's and world's accelerator is LHC – Large Hadron Collider, the circumference of this incredibly complex machine equals 27 km. The LHC beam feeds the main experiments such as ALICE, ATLAS or CMS. The NA61/SHINE works with a beam from SPS which is a much smaller accelerator with a circumference of 6,9 km, SPS provides the beam to LHC.

1.1.1 Information technologies at CERN

The Particle Physics and the Computer Science share the common areas. For instance, the need of collecting a terra bytes of data within a strict time constraints as well as performing later data analysis are a very interesting and challenging fields of the Computer Science. These two topics are very important for high-energy physic's experiments, however it's not the only space where the Information Technologies can be applied, an excellent example is WWW.

World Wide Web

The most important contribution of CERN to IT world and very important step in development of a network technologies is the World Wide Web – in short known as WWW or Web. The scientific community involved to CERN experiments is distributed all over the globe, the demand for intuitive and universal data sharing system was particularly important in this circles.

WWW was released in 1991 by Tim Berners-Lee who was a CERN employee at that time, within a short period the system gained in a popularity. Nowadays WWW is so popular that is often mistakenly confused with the term “internet”.

1 number of active users in year 2011

2 the abbreviation CERN stands for French: Conseil Européen pour la Recherche Nucléaire, this name was in use for a short time period until this organization officially came into being in 1954, despite changing the name the old abbreviation was preserved

GRID

CERN currently develops the GRID system that provides the distributed data storage and computing power. These two factors are very important in this field of science as the data and data processing requires enormous amounts of resources. As the WWW spread over the world because the data sharing system was desired in practically all fields, the GRID is desired outside of particle physics' world too, in disciplines such as biology or climatology where the computer simulations are essential and are definitely a benefice of GRID.

Scientific Linux

Linux as a leading server operating system is represented by hundreds of distributions where each of those has a place in a computer world. CERN in cooperation with Fermi National Accelerator Laboratory (Fermilab) and various other laboratories develops and maintains Scientific Linux operating system based on Red Hat Enterprise Linux. The system is aimed to be used in the research centers, and to be a common install base for the various experimenters.

1.2 CERN Technology Transfer

The term technology transfer is the process of moving the technological knowledge to another organizations or companies. CERN is very successful in this field, mentioned World Wide Web is the most striking example of CERN's technology transfer. Another example is the GRID that is already gaining in popularity being implemented to new fields.

The Computer Science as it was mentioned in point 1.1.1 is very strongly connected with Particle Physics by the need of dealing with huge amounts of data. However CERN is not only about data, the technologies developed for accelerators and detectors found an important space in the medicine. The example of accelerator's technologies shifted to that field is Hadron Therapy, The Hadron Therapy is used for tumors treatment, this technology is based on pioneering studies that were carried out at CERN in late '60s. Another example are isotopes, nowadays most of them are produced in nuclear reactors, however the new studies are carried on production of new types of isotopes using particle accelerators. The development of detector technologies brings to medicine new medical imaging systems. Just one of the examples is Medipix2 which adopted the CERN's research to a new X-ray imaging system that eliminates the background noise which is an issue of traditional X-ray approaches.

The Energetics industry can also benefit from Technology Transfer, the ultra-high vacuum technologies were applied to produce the prototypes of evacuated flat panel solar collectors that can utilize the sun's energy in a more efficient way.

It's worth to mention that CERN supports knowledge exchange across the electronic design community, on March 2011 CERN issued version 1.0 of „Open Hardware Licence” and on July 2011 version 1.1. This licence implements the open source concept very well known from IT to an electronic industry. The Open Source nowadays plays an important role, the Open Hardware Licence, or simply Open Hardware is a chance to bring new designs that can be shared and improved by the community.

A good source of information about CERN's Technology Knowledge is [CERN - Knowledge].

1.3 NA61/SHINE experiment

The NA61/SHINE is one of the smallest CERN's experiments, however despite the size the scientific contribution is definitely significant. In total in project 27 institutes (sometimes referred as a groups) takes a part, among them seven are Polish. About 200 persons from fifteen different countries contributes to the experiment.

The abbreviation NA61 denotes „North Area 61” what indicates experiment number 61 located in the CERN's North Area. SHINE stands for “SPS Heavy Ion and Neutrino Experiment” what points out the purpose of the experiment which is to investigate heavy ions and neutrinos using SPS beam. The acronyms NA61 and SHINE can be used interchangeably, however SHINE as a new name is preferred, the NA61/SHINE is used too as it binds old and new naming.

1.3.1 NA61/SHINE Science

The main aim of the experiment is to find critical point of strongly interacting matter. In a simple way the critical point can be described as an energy level at which the strong interactions that binds the structure of atom's nuclei are severed, thereafter the matter changes to so called quark-gluon plasma state. In SHINE the quark-gluon plasma is produced by colliding a particles previously accelerated with a target. For example the possible collisions are the collisions of a lead ions with a lead target, this kind of interactions is shortly called PbPb. The experiment's aim can be illustrated by an analogy to phase diagram of water, where the different phases depends on the temperature and pressure. The NA61/SHINE experiment aims to provide data that will allow physicists to develop an analogical chart for the matter.

The products of collisions are registered by a set of sub-detectors, the data collected in this way allows on reconstruction of particle's tracks and particle identification. The NA61 detector is about 4 meters high and 13 meters length, comparing to another CERN's detectors the NA61 is relatively small.

It's worth to mention that SHINE cooperates with T2K – Tokai to Kamioka experiment that is hosted by JPARC – Japan Proton Accelerator Research Complex. The NA61/SHINE detector data will help to develop better model for describing the production of pion and kaon particles in collisions of protons with a carbon target. This model is an essential for understanding the data from T2K.

1.3.2 Work in NA61/SHINE

One of the terms typical for partial physics experiments is „Run”. It is the time period when the particle physics detector collects data or is under calibration in order to achieve the best measurements results.

Typically during the year, two runs are performed. The first one is called a „test run” which usually takes place during the spring time. This run lasts for about a week or two, it's goal is to test and calibrate hardware and software before a the second one – the „physics run”. The physics run usually starts a few weeks later and lasts much longer, in general it takes about a few months. This run is intended for a data collecting.

During the run time the detector must be continuously monitored. This necessity obligates NA61 collaborating institutes to designate members to stay in the experimental area in order to watch and keep a detector running. The time period when a member is obligated to stay is called a shift, the person that is working during this time is called shifter.

1.4 My contribution to NA61/SHINE

In NA61/SHINE experiment where many institutes from around the world are involved, the need to organize the work in an efficient way led to develop a set of dedicated software applications.

1.4.1 Software requirements

The requirement of easy access to the applications over internet determined the choice of WWW technology as unrivalled. An important features that draw the software development were:

- scalability – it's a ability to maintain constant application performance regardless of amount of data stored by the system. It's desire because the system will be collecting data over a years.
- flexibility – an application featured by this property is easy to be adopted to the new requirements. The software that is intended to support a work of a big and a dynamic group should be prepared for an unexpected changes.
- security – means to protect the data against unauthorised access. This factor is very important as the applications are accessible directly from the internet.

Part of my master thesis are applications to support administrative tasks as well as scientific data analysis. The first application is the *ShiftsScheduler* which helps to schedule the shifts for upcoming runs. The next application is *ElogLooker* that allows on data analysis.

1.4.2 NA61/SHINE work organisation with ShiftsScheduler

Most of scientists involved in experiment on everyday basis doesn't works at CERN but in theirs home institutes. For the run time the institutes delegates employees involved in NA61 to perform necessary work at the CERN side.

As it was mentioned in point 1.3.2 during the run time the detector must be continuously watched, in addition – according to security rules at least two persons must be present at detector area. It's advised to have one experienced and one non-experienced person, so the knowledge can be transferred.

Necessity of a shifts planning in an efficient way created a space for a shifts scheduling software. The Shifts Scheduler application implements mentioned requirements, but as well brings a series of an additional features such as run statistics or contact book.

1.4.3 Run statistics with ElogLooker

During the run time so called “run logbook” is maintained where detector parameters supplemented by optional comments can be noted.

In the past the standard paper notebook was used, however NA61 waived usage of paper in the favour of electronic record. Software chose by collaboration for logbook keeping is called “Elog”.

1.4.3.1 Elog and ElogLooker

Elog is a network software developed at the Paul Scherrer Institute in Switzerland. The program works as a server, clients can establish connection by using just WWW browser. This software is very suitable to share logbook data over a group of people, the server is easy to install and configuration, for more information please refer to [PSI - ELOG usage] or [PSI - ELOG administration].

Usage of a computer version of logbook opens a new qualities, it become possible to automatically generate the statistical data based on logbook entries. However, introduced Elog software is not capable of this functionality. Here comes the next application developed by me – ElogLooker.

Whenever new entry appears in Elog, the ElogLooker is automatically notified about the change, thus updated statistics can be imminently generated.

2 Requirements for applications

This chapter introduces the definitions and requirements that are the foundations of *ShiftsScheduler* and *ElogLooker*. The applications functionality is a derivative of these factors, which at the end shapes the user's interface. This chapter and a chapter 4 – User's interface are complementary in terms of the application description from the user's point of view. The reader who is not interested in technical details can read these two chapters in order to have a good overview on described services.

2.1 ShiftsScheduler

The point below describes the basics of the *ShiftsScheduler* application, The application is intended for shifters and for group leaders. Even though a group leader is not shifting, he/she is still responsible for shifts organisation within his/hers group, so they should have an access to this service.

2.1.1 Shifts

Each shift is 8 hours long, due to the security reasons it's obligatory that two shifters must be present at the same shift. Depending on at what time of day does the shift takes place is called as follows:

- night shifts – takes place from 0:15 to 8:15
- day shifts – takes place from 8:15 to 16:15
- evening shifts – takes place from 16:15 to 0:15 (at the next day)

Each 24 hours period is divided by three shifts and for each of the shifts two places for two shifters are presented, see table 1.

night shift	day shift	evening shift
Shifter 1	Shifter 1	Shifter 1
Shifter 2	Shifter 2	Shifter 2

Table 1: representation of the shifts and shifters in a 24 hours period

As you can notice, in table 1 each column represents the shifts in the order of occurrence, e.g. night shift, day shift, evening shift. Column is split into two cells where the first one is over the second one. These two cells represents two shifters that are obligate to be during the shift time.

This representation is used later in the system, you can find more details about usability of Shifts Scheduler in chapter 4.1.

2.1.1.1 Shifts reservations and confirmations

The shifts signing up process is divided into reservation and confirmation.

The shifts assignment consists of two parts, first shifter is making a reservation for the time period when he/she wants to work, secondly the group leader or administrator has to confirm it.

A reserved shift can be cancelled by a user. When the shift is confirmed by the group leader, only the group leader or administrator can cancel it. However there are exceptions which is described in the next points.

2.1.2 Account types

The Shifts Scheduler provides three types of user's accounts which are assigned for users depending on their duties and role during the run time. These accounts are as follows:

- shifter's account
- group's leader account
- administrator's account

Depending on the account's type users have the possibility to do a certain actions or not. In general all account types can be used to make reservations, however group's leader account has an extended capabilities over shifters account while administrator's account has an extended capabilities over group's leader account. Depending on the account type user can do the following actions:

- making and cancelling reservations
- confirming and rejecting reservations
- adding and removing users accounts
- editing profile

The Table 2 shows the privileges associated with the certain account types for the certain actions. For instance the group leaders can reserve shifts only for theirs group members while the administrator have the possibility to make a reservations in charge of all members from any group.

As the table shows the reservation can be done from any account type. Regarding confirmation the account of a group leader or administrator must be used. When the reservation is once confirmed, the only person who is in power to cancel it is a system administrator.

2.1.3 Reservations on behalf of another person's

The **group leaders** and **administrators** are in power to make actions on behalf of other shifters.

Making an action on behalf of somebody means that the action will be accounted as it could be done by this very person, however the system will add a notice about the person who did the action.

In Shifts Scheduler two actions can be done on „behalf of”, these are:

- making reservations
- cancelling reservations

Account type	Action	Can the user with a certain account type do the following action in charge of ...		
		... himself ?	... somebody else from his group?	... somebody else from any group?
shifter	(making reservations / cancelling reservations) ³	yes/yes	no/no	no/no
group's leader		yes/yes	yes/yes ⁴	no/no
administrator		yes/yes	yes/yes ⁴	yes/yes ⁴
shifter	confirming reservations / rejecting reservations	no/no	no/no	no/no
group's leader		yes/no	yes/yes ⁴	no/no
administrator		yes/no	yes/yes ⁴	yes/yes ⁴
shifter	cancelling confirmed reservations ⁵	no	no	no
group's leader		no	no	no
administrator		yes	yes	yes
shifter	adding or removing users accounts	no	no	no
group's leader		no	yes	no
administrator		no	yes	yes
shifter	editing profile	yes	no	no
group's leader		yes	no	no
administrator		yes	no	no

Table 2: Account types and privileges

According to the Table 2, group leaders can make reservations on behalf of members of his group, while administrators can do this action on behalf of all shifts scheduler users.

Users that are allowed to do actions on “behalf of” other shifter must indicate in the Shifts scheduler that certain action should be done on behalf, for more details go to point 4.1.1 – Actions on “behalf of” other shifters.

³ these actions can be done on behalf of other users, for more details see section 2.1.3 - Reservations on behalf of another person's

⁴ the difference between shifts cancelations and rejections is explained in the point 2.1.4 - Reservation cancelling vs. rejecting

⁵ to simplify the terminology this document later refers to “confirmed reservations” as “confirmations”

2.1.4 Reservation cancelling vs. rejecting

This point describes details between reservation cancelling and rejecting. Both operations leads to the same result – the reservation is removed and the free place is ready to be taken again.

The different terminology was introduced to emphasis from what point of view the operation is performed.

If the reservation was made by you, but due to some reason you want to retract from your decision, you **cancel** your reservation. However if somebody else who has sufficient privileges (most likely your group leader) doesn't agree upon your decision about reservation, he doesn't cancel it but **rejects** it.

In the case of reservations made on behalf of other users where two persons are involved, e.g. group leader and shifter, the terminology is as follows:

When one of these two persons wants to discard the reservation the **cancel** option must be used. Please note that if a third person is involved who wants to discard that reservation (e.g. administrator), but this person was not involved in booking an reservation, then he must do **reject** an reservation.

2.1.5 Automatic confirmations

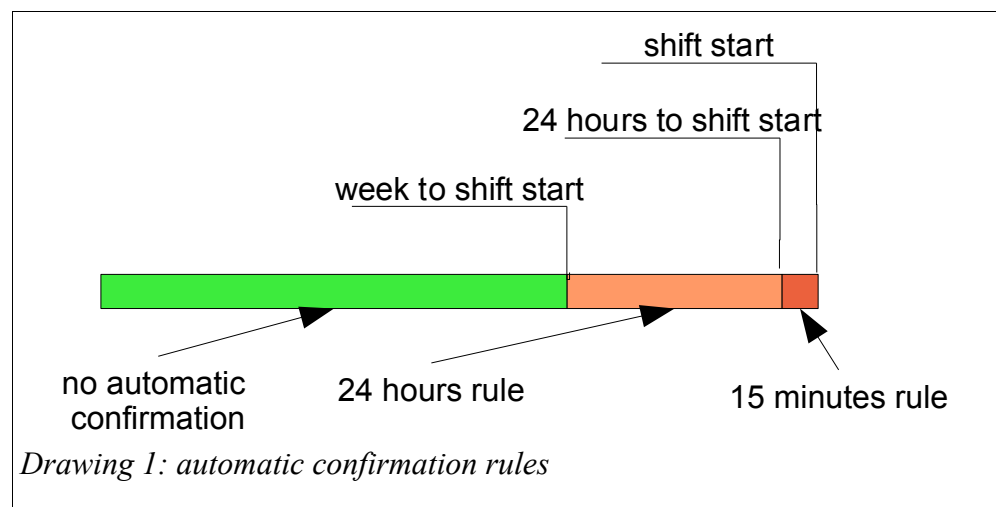
In the case when the reservation has been done but not confirmed before one week to start, the shift will be confirmed automatically by the system. As in the case of human confirmation user is obligated to come to contribute in the shift.

The automatic confirmation rules are as follows:

- If less then a week but more then 24 hours to shift start remains the 24 hours rule is applied.
This rule states that if no manual confirmation is make within 24 hours the shift is confirmed automatically after that time.
- If less then 24 hours to shift start remains the 15 minutes rule is applied.

Analogically to the previous rule the shift is confirmed automatically if no manual confirmation was made but after 15 minutes.

The automatic confirmation rules are shown at the Drawing 1. As an example, assuming that no person will confirm the reservation, the reservation made in more then a week to shift start (green bar) will be automatically confirmed in a six days before start. That's because as the time pass by at one point the remaining time to shift start will be one week, then the 24 hour rule applies (orange bar), so the reservation will be confirmed after 24 hours what means six days before start.



As you can notice the reservation can be booked at the very last time, e.g. in 15 minutes before run start. This situation shouldn't happened, but if it's so the reservation will be confirmed immediately.

It's a duty of a run coordinator to avoid situation when the shift goes without one of the shifters what is an unacceptable situation. If it would happened the shifts scheduler will mark such a shift in red adding message describing situation.

2.1.6 Blocs

The previous paragraphs has explained the most important rules of the Shifts Scheduling application such as account types and the shifts reservations.

The simplest scenario for shifts booking system is when one reservation points exactly one day and one shift, e.g. 1st of July, the night shift. However, this approach for a long runs like the ones that remains for a few months, causes a confusion and makes the shifts planning and managing more difficult mostly because of an inconsistency⁶ of reservations.

To addresses this issue, so called „block” reservations has been introduced. The blocks covers a few days in a row, it's not permitted to reserve a shift for one day only but the shifts for days which block covers must be reserved. The block is usually three days long, but for some rare cases blocks might cover just one day.

To be more precise, the blocks covers not only a days, but the shifts, e.g: night, day or evening shifts. It means that by reserving e.g. night shifts blocks, implicates reservation of night shifts for a days that are covered by the certain block.

2.1.6.1 NA61 blocks layout

If the blocks are longer, the system is easier to manage, but shifts choosing flexibility suffers. However the blocks layout implemented in NA61 preserves pretty much flexibility, and still ensures that the system is quite good to manage. This solution is presented as an example in Drawing 2.

⁶ inconsistency in terms of reservations means that shifts reserved by a certain person or group are distributed over a long time period, so in between the shifts there is a time period which is not reserved by this person or group, this situation is not convenient for a groups which are traveling from outside of Geneva region (what concerns most of the groups) particularly for shifting.

The Drawing 2 shows an exemplary six days of a run, the day consists of a three shifts with two places per shift for shifters as it was explained in the Table 1 of paragraph 2.1.1 – Shifts.

	night shift	day shift	evening shift		night shift	day shift	evening shift		night shift	day shift	evening shift
Shifter 1:	B 5	E 1	2		5	B 7	E 2		E 5	7	B 9
Shifter 2:	3	B 6	E 4		E 3	3	B 8		B10	E 6	8
	day 1				day 2				day 3		
Shifter 1:	B11	E 7	9		11	B13	E 9		E11	13	B15
Shifter 2:	10	B12	E 8		E10	12	B14		B16	E12	14
	day 4				day 5				day 6		

Drawing 2: an example of a blocks of shifts,
B x – beginning (the first day) of a block *x*
E x – end (the last day) of a block *x*

Each block for this example is a three days long, the first day of a block is indicated by a letter 'B' followed by a unique number which identifies the block. The last day of a block is indicated by an 'E' letter followed by a block number. The second day of a block is indicated by just a number.

To improve the readability of example the blocks that starts on the same day are indicated by the same colour.

For instance the block B5 starts from day 1 and lasts until day 3. As the Drawing 2 indicates it covers the night shifts for a shifter 1 over this three days. Shifter by reserving this block reserves automatically these three shifts.

Please notice that the blocks 1, 2, 3 and 4 started before day 1 while the blocks 13, 14, 15 and 16 I'll end after day 6,

There are two important advantages of a shown layout:

- every day two new blocks starts⁷
- two new blocks that starts each day starts at the different time

With thanks to this layout the Shifts Scheduler is pretty flexible, but the clarity of assignments is still preserved.

The blocks size doesn't exceeds three days, however shifters are encouraged to take two blocks in a row, what makes the shifts scheduler's calendar even more clear.

The way how does the shifters are motivated to shift two blocks in a row is presented in the next paragraph.

⁷ the exception are usually the first days of a run when all blocks must start

2.1.7 Points awarding

The NA61 collaboration uses points to assess the contribution of an institute as well as individuals during the run. The shifters by reserving the blocks are collecting the points. The points granted for an institute are the sum of points of shifters from this institute.

The institute is obligated to gain at least the amount of points calculated as follows:

$$| \text{minimum amount of points for institute} = 6 * \text{quantity of authors from institute}$$

The institutes must reach at least the amount of points equal to quantity of authors multiplied by six, points are accounted annually. The beginning of the year all institutes start with zero balance, the first opportunity to collect points is during the first run – usually it's a test run. The points from different runs are summarized. The minimum amount of points have to be obtained or exceeded until the end of the last run of the year.

The amount of points is calculated in terms of quantity of taken shifts as well as in terms of shift difficulties. By the shifts difficulties we understand the factors that make a certain shift or shifts inconvenient, these factors are:

1. the night shifts (between 0:15 and 8:15) - implicates the rule 1, see section 2.1.7.1
2. taking two blocks in a row what means for the most cases to take six days in a row - implicates rule 2, see section 2.1.7.1

2.1.7.1 Points awarding rules

In general one point is awarded for one shift, however the system takes into account mentioned inconvenience factors, so the points can't be counted in such a straight way. Amount of points granted for a certain block is calculated according to three rules explained below. This point refers to the examples that can be found in the next paragraph: 2.1.7.2 – Points awarding examples.

Rule 0

Applies when block is considered as convenient:

$$| \text{ amount of points for one block} = \text{block length} - 0.5$$

The rules 1 and 2 works when the block is considered as inconvenient, hence more points is granted. The examples 1 and 3 are referring to this rule.

Rule 1

Applies for the night block:

$$| \text{ amount of points for one night block} = \text{block length}$$

By taking the night block half of point is given extra. The example 3 is referring to this rule.

Rule 2

Applies for the block that is directly preceded by exactly one block reserved by the same person:

$$| \begin{aligned} &\text{amount of points for a block directly preceded by exactly one block reserved by the same person} \\ &= \text{block length} + 0.5 \end{aligned}$$

The block is directly preceded when the next block starts right at the next day after the previous one ends, so there is no single day that has no shift reserved – in another words the blocks are in the row.

As you can see, by taking the second block right after the previous one, one extra point is given comparing to the block that is booked “alone”, see example 1.

Please notice that the rule 1 and 2 can apply at the same time. It happens when the block that is directly preceded by exactly one block reserved by the same person is in addition the night block:

$$| \begin{aligned} &\text{amount of points for a night block directly preceded by exactly one block reserved by the same} \\ &\text{person} = \text{block length} + 1 \end{aligned}$$

In this case an additional one and a half extra points is awarded comparing to the block where the rule 0 only applies – example 4 refers to this case.

Two blocks limit

Shifter shouldn't work longer then six days without having a day off. That means that no more then two 3 days long blocks can be taken in a row. Shifting for longer time is not forbidden, however it's not recommended.

Due to this reason, in the case when more then two blocks have been taken in a row, the blocks that appears after the second one are not granted with any bonus points, see Example 6.

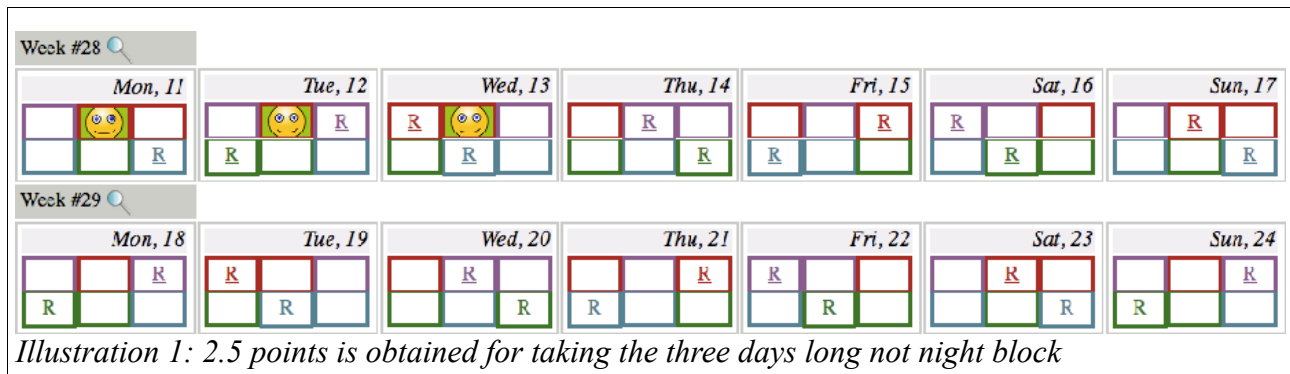
2.1.7.2 Points awarding examples

Presented examples refers to the future point 4.1 in chapter „User's interface“ where the colours and iconography are explained, however at this point it's sufficient to know that the face icon represents the shifts reserved by a shifter while the 'R' letter means the beginning of the block.

Example 1 – rule 0

The block is not the night block and it's not directly preceded by exactly one block, so the only rule 0 applies:

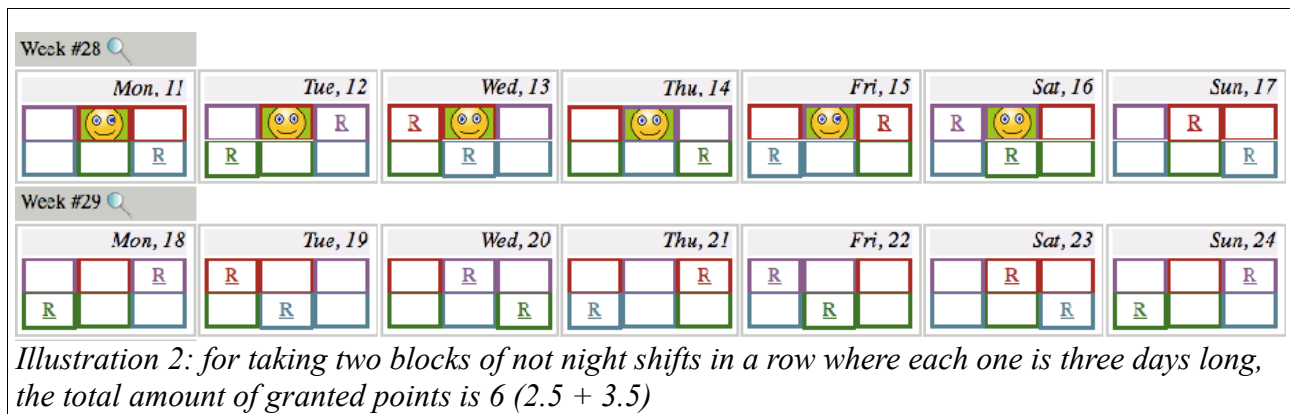
$$| \text{ points} = \text{block length} - 0.5$$



Example 2 – rule 0 and 2

If the second block is directly preceded by another one (the second one starts on the next day when the previous ends) a rule 0 for the first and rule 2 for the second block applies:

$$| \text{ points} = (\text{first block length} - 0.5) + (\text{second block length} + 0.5)$$



Example 3 – rule 0

If the blocks are not in a row rule 2 can't apply so two times rule 0 applies:

$$| \text{ points} = (\text{first block length} - 0.5) + (\text{second block length} - 0.5)$$

Week #34						
Mon, 22	Tue, 23	Wed, 24	Thu, 25	Fri, 26	Sat, 27	Sun, 28
Week #35						
Mon, 29	Tue, 30	Wed, 31				

Illustration 3: in the case of brake between two three days long not night blocks 5 points is awarded (2.5 + 2.5)

Example 4 – rule 1

If taken block is the night block, then the rule 1 applies:

$$| \text{ points} = \text{block length}$$

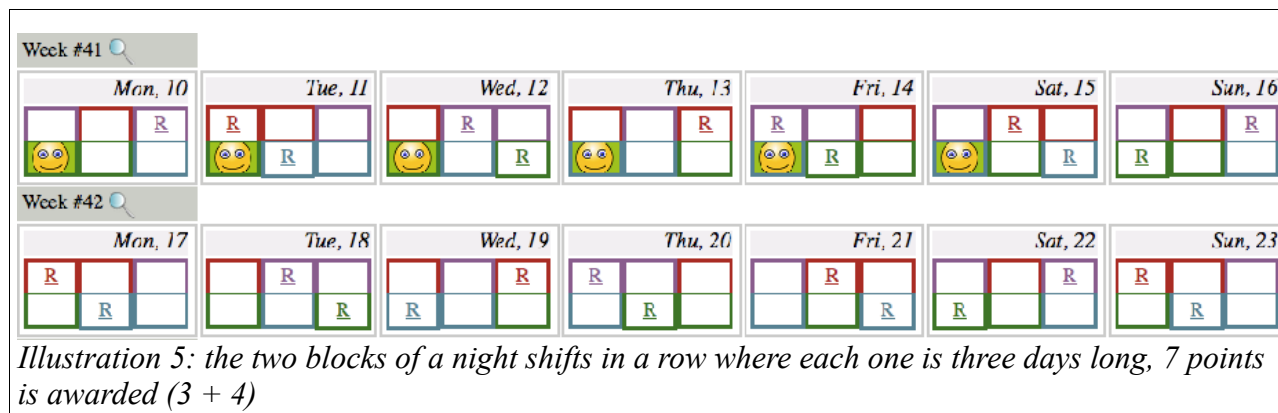
Week #41						
Mon, 10	Tue, 11	Wed, 12	Thu, 13	Fri, 14	Sat, 15	Sun, 16
Week #42						
Mon, 17	Tue, 18	Wed, 19	Thu, 20	Fri, 21	Sat, 22	Sun, 23

Illustration 4: the three days long night block is rated at 3 points

Example 5 – rule 1 and 2

If there are two night blocks in a row then the rule 1 and 2 applies:

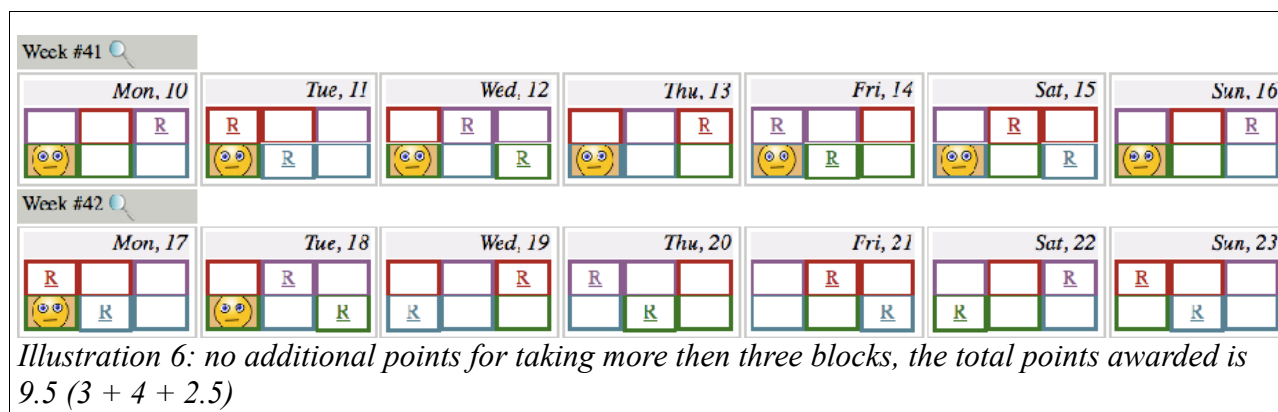
$$| \text{ points} = (\text{first block length}) + (\text{second block length} + 1)$$



Example 6 – rule 0, 1 and 2

For taking three night blocks in a row no any bonus points for a third block is awarded. For the first block the rule 1 applies, for second 1 and 2 but for third just rules 0.

$$| \text{ points} = (\text{first block length}) + (\text{second block length} + 1) + (\text{third block length} - 0.5)$$



2.2 ElogLooker

The *ElogLooker* is a very small application comparing to *ShiftsScheduler*. As it was introduced in point 1.4.3 – „Run statistics with ElogLooker“ the program complements the Elog by adding the data statistics capabilities.

Elog is designed to store data in the way that is known from computer world as data logging. The data logging it's just saving the events with a time stamp that occurred during the system run for a future analysis. In *elog* any entry is characterized by a time stamp and a certain – chosen by administrator data fields. The entry is added within a logbook, the *Elog* can hold many separated identified by a unique name logbooks.

The NA61/SHINE collaboration uses several logbooks for keeping all records that the experiment requires. However, at this state the statistics are generated for only one logbook that is called „Run summary“ which is used to store run reports. The statistics that the application generates are very simple, it's just a sum and a count of events for a certain beam type⁸ and a certain beam energy⁹.

Although application may look simple, the generic approach makes it very configurable and easy for adaptation to new requirements, e.g. attached to a new logbook. However this approach makes software more difficult to develop.

⁸ beam type determinates what kind of particles are collided in detector

⁹ beam energy given in GeV/c

3 Used technologies

Since the first release of World Wide Web in 1991 the capabilities of this technology evolved from a simple data sharing to an advanced network application services. The web applications are very convenient from users point of view as to use it only web browser is required. It's particularly valuable when the application is designed to exchange information between users like *ShiftsScheduler* or *Elog*, in this case the server provides both: the application and data.

The web application can be structured into three tiers:

- **Presentation tier**, the presentation in this terminology is a user interface developed in HTML and related languages, this tier belongs to the client's side as the web page is rendered by a client's web browser.
- **Application tier**, the application tier implements an application logic like users validation or privilege control. This part runs at the server side, in most cases it is developed in programming languages like Java, C# or C/C++¹⁰ - what is in minority these days.
- **Storage tier**, in the storage tier the data are stored. The application server requests data from database using a special interface, so the internal structure of database and database type are separated. It makes system more flexible as a change in database structure or the change of database to another type requires only change in the storage layer.

The web application is a mixture of a different technologies where each of those is the most suitable for a certain tier. This chapter describes technologies used to develop *ShiftsScheduler* and *ElogLooker* basing on three tiers classification.

3.1 WWW – presentation tier

The World Wide Web is a set of three essential technologies:

- **URL** – Uniform Resource Locator
- **HTML** – HyperText Markup Language
- **HTTP** – HyperText Transfer Protocol

These technologies are related each other in the way where URL address identifies the HTML document that is published at the WWW server, the document can be accessed by a web browser via HTTP protocol. The presented solutions are standardised by the World Wide Web Consortium (W3C). The next points focuses on detailed introduction to these standards.

¹⁰ WWW server can pass HTTP requests to environment variables using CGI (Common Gateway Interface), the values from environment variables can be read by a program written in any language, thus making possible to apply C/C++ programs to deliver HTTP response

3.1.1 URL

Uniform Resource Locator or Universal Resource Locator is the address of a documents and other resources on the WWW. URL combines existing since 1985 domain name system with file path syntax and transfer protocol.

According to [Berners-Lee, T.; Masinter, L.; McCahill, M. - URL] the syntax of URL is as follows:

| *scheme://domain:port/path?query_string#fragment_id*

The semantics of each field is as follows:

scheme name	– determinates the url protocol, in the most cases the HTTP (see point 3.1.3) protocol is used, another frequently used protocols in URL addresses are: – ftp, File Transfer protocol – https, Hypertext Transfer Protocol Secure – mailto, Electronic mail address
domain name	– name registered in a Domain Name System (DNS)
port number	– optional port number, if it's not given the default port for chosen protocol in scheme name is used, eg. for HTTP port 80 is used
resource path	– path to resource at the server that is indicated by a domain name, it's a file path syntax where forward slashes are used to separate folder and file names
query string	– a string that contains data to be passed to web application, data are given in a format: field1=value1&field2=value2... sign '&' is used as separator
fragment identifier	– optional identifier to specify a part within resource or a position in a document

An example url from *ShiftsScheduler* application is shown bellow:

| *https://na61db.cern.ch:8443/ShiftsScheduler/weekly_view.xml#week46*

The https indicates secure version of http protocol, na61db.cern.ch is the name of the server where application works, 8443 indicates a port number that is not a standard one as a default for https is 443. Path */ShiftsScheduler/weekly_view.xml* indicates XML document, reference *#week46* points position in *weekly_view.xml* document where information about week46 is given.

3.1.2 HTML

In order to describe the web based technologies it's important to explain meaning of the abbreviation HTML – HyperText Markup Language. This phrase can be split into two parts: HyperText and Markup Language.

HyperText

Hypertext according to [McGraw-Hill - Directory] is:

“A data structure in which there are links between words, phrases, graphics, or other elements and associated information so that selection of a key object can activate a linkage and reveal the information”

The hypertext concept existed a well before HTML was developed, by simplifying this definition in the context of HTML, the new definition is as follows:

“The HyperText is the text where the references (hyperlinks) to complementary text or media (e.g. pictures, movies) are embed, the linked text or media can be easily accessed usually by a single computer mouse click¹¹”

For a web software it become a standard to present a hyperlinks – or simply links as a underlined blue in colour text. To make a system more intuitive a mouse cursor when it's over the hyperlink changes from arrow to pointing hand icon.

Markup Language

The Markup Language is:

“a modern system for annotating a text in a way that is syntactically distinguishable from that text”

It means the system where an additional text can be used in order to mark that a certain text is characterised by a some properties. These properties can specify how to present a text, e.g. size of a font, or what kind of data the text represents, eg. document title. In HTML a text is marked with so called 'tags'. Semantics of tags in contrast to XML language (described later in point 3.1.2.4) are predefined by HTML standard.

HTML standard

The HTML as the sentence „HyperText Markup Language” indicates is a Markup language designed for describing the HyperText documents.

The HTML document is defined by tags and it's content, the structure of a tag element according to [W3C - HTML] is presented at the Illustration 7.

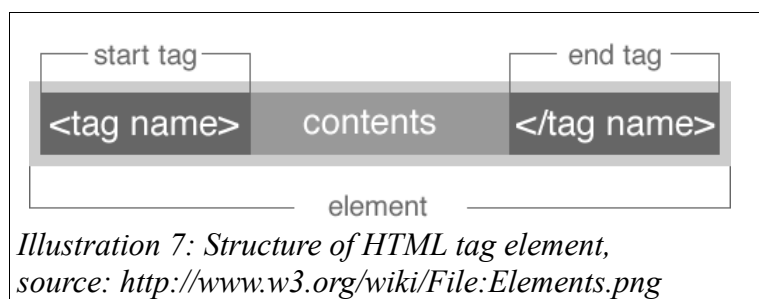


Illustration 7: Structure of HTML tag element, source: <http://www.w3.org/wiki/File:Elements.png>

¹¹ in some rare cases an equivalent of a single click can be specified by a keyboard key (like enter in lynx web browser)

The tag defines how does the content of tag should be interpreted, e.g. tag *table* indicates that included content should be interpreted as a table. An additional tag arguments can be passed to tag in order to define more strictly the behavior of tag, e.g. to specify the width of the borders around a table. The tag attributes syntax is as follows:

```
| <tagname attribute1="value1" ...></tagname>
```

Please note that that the content of tag is surrounded by an opening tag *<tag name>* and closing tag *</tag name>*. However some tags are so called empty tags, in this case there is no closing element as well as there is no tag content to mark. One of the examples of empty tags is “
” that defines the line break in text.

The basic structure of HTML document is as follows:

```
| <html>
  <head>
    <title>The document title</title>
  </head>
  <body>
    <h1>Main heading</h1>
    <p>A paragraph.</p>
  </body>
</html>
```

Tag html is so called root element, it can be exactly one defined in document. HTML is split to head and body parts, respectively represent by head and body tags. “head” tag is a container for any information that is not considered as a document content, this information may be a title (shown in an example) or keywords that are useful to search engines. “body” is the element where all information about document content is included, such as shown in the example definitions of header and paragraph.

The HTML language is case insensitive, so the tags and attributes names can be written without paying an attention to characters case, however it is considered good practice to keep HTML mark-up lowercase.

“A” tag

The crucial functionality of HTML is HyperText, the hyperlinks to another documents can be defined by the 'a' tag, the basic syntax of this tag is as follows:

```
| <a href="URL">URL description</a>
```

The URL address where the hyperlink refers is pointed by a value of *href* attribute. Any text that surrounded by open tag and close tag will be interpreted as a link name and by default presented in [blue underlined text](#).

3.1.2.1 CSS

Good web developer practise is to separate the web page structure of from presentation semantics – look and a formatting of document.

Cascading Style Sheets (CSS) is a style sheet language designed to describe how does the HTML document should be presented. For instance it can be used to set a font style or background colour.

CSS styles informations can be placed in the following locations:

- | | | |
|-----------------------|---|---|
| inline styles | – | inside the HTML tag using „style” attribute |
| embedded style | – | block of CSS styles definitions inside HTML document |
| external style sheets | – | separated file that contains exclusively CSS styles definitions. In order to include CSS definitions from a separated file to HTML document the „link” tag with URL do that CSS file must be included to the document |

For each location the priority is associated, for the inline style the highest priority is given, for embedded style and external style sheets respectively - lower and the lowest priorities are assigned.

If a certain style applies in inline styles and external style sheet then the inline style applies, the style from an external style sheet can apply only when the first one is removed. If a certain style is not defined at all, then the default web browser's style is used.

If one HTML document includes many external style sheets files the file that was included as last (with „link” tag) has the highest priority.

Word 'Cascading' from CSS abbreviation indicates that the system in order to define from which place what style should apply to the document follows the priorities, this process is called „cascading”.

CSS syntax

The CSS syntax is very different then HTML, thus it is not too confusing with HTML. The basic syntax consists of only three parts:

```
selector [, selector2, ...] [:pseudo-class]
{
    property: value;
    [property2: value2;
    ...]
}
/* comment */
```

The selector is a HTML tag that is going to be styled, pseudo-class is used to identify markup elements more precisely, for instance pseudo-class first-line applies only to a first line of a text. The property is a style property like margin or font size while value is the value of this property, e.g. „12pt” can be used as a value for font-size property. The CSS is standardized by W3C, the specification of CSS 2.1 that is currently recommended can be found in [W3C - CSS].

3.1.2.2 ECMAScript

ECMAScript is a scripting language designed to use on web on the client-side. This tool makes a possibility of creating more interactive web pages without action from the server side as the ECMAScript code is executed on the client side by a web browser.

JavaScript, JScript and ActionScript

ECMAScript is standardised by Ecma international organisation in the ECMA-262 specification [Ecma - ECMAScript]. The history of ECMAScript starts with scripting language developed by Netscape Communications Corporation under the name Mocha, it was renamed later to LiveScript and finally to JavaScript. In March 1996 the Netscape Navigator 2.0 web browser – the product of Netscape was featuring JavaScript. Due to success of JavaScript Microsoft implemented compatible language called JScript in Internet Explorer 3.0 in August 1996.

The name JavaScript is confusing as it refers to Java developed by Sun Microsystems (see point 3.2 – Java – application tier) while there is a very little similarities. Referring to [Champeon, Steve - JavaScript] JavaScript was called „complement” to both Java and HTML, word „Java” might be used for marketing reasons as Sun's Java was very promising uprising platform on that days.

On the same year Netscape committed JavaScript to Ecma international for standardisation, the first edition of ECMA-262 was published in June 1997. The name ECMAScript is a compromise mostly between Microsoft and Netscape.

Nowadays in addition to JavaScript and JScript a set of new ECMAScript based languages were developed, one of thus is ActionScript by Macromedia inc.. All that languages are called a dialects, a dialect is a superset of the syntax and semantics of ECMAScript language. The ECMA compliant scripts works correctly with dialects that are based on proper edition of ECMA standard, however if a script is extended by a dialect specific features it may be not compatible with another dialects.

The name JavaScript become so common that very often is mistakenly used as an equivalent of ECMAScript. The programmer's guide and reference manual for JavaScript can be found in [Flanagan, David - JavaScript].

3.1.2.3 DOM model

Document Object Model (DOM) is a convention of representing HTML, XHTML or XML elements as a objects that can be interactively modified. The benefit of DOM is a possibility of accessing and modifying tags in HTML document with scripts like introduced ECMAScript what leads the way to more interactive webpages.

One of the examples of DOM application would be a web page where the background can be choose by a mouse click on the button element. In pre DOM and pre JavaScript times for such an interaction the action from the server side would be required. Nowadays DOM could be used as a interface to access appropriate HTML elements such as 'buttons' that has been used for color selection or the element where the background should be set, while the scripting language would act as binding part where each button would be associated to a certain color, e.g. with switch statement ¹².

¹² switch statement in computer programming is a type of selection mechanism, programmer using switch statement can define what instructions should be executed depending on a values given to the statement

3.1.2.4 XML

XML - eXtensible Markup Language, it's a language aimed to transport and store data, it's flexibility in defining the data structure suited XML as a very successful standard.

XML is a W3C recommendation, the specification can be found in: [W3C; XML]. XML uses well known from HTML language tags to mark-up the data. The semantics of tags is defined by software/protocol designer, as an example of XML code I attached one of the configuration files from the program for remote computers control which I developed as a part of my bachelors thesis [Piwek, Mateusz - kontrola komputerów].

```
<?xml version="1.0"?>
<!DOCTYPE body [
  <!ELEMENT body (tool_declaration*, tool_instance+, notification*, task_mask?)>
]>
<body>
  <tool_declaration name = "tool name" alias = "tool alias"/>
  <tool_instance alias = "tool name alias" name = "instance name" class = "class name"/>
  <notification>
    <!-- determinates way to send notifications and recipients of reports -->
  </notification>
</body>
```

„<!ELEMENT” is a declaration of the data structure, it's an XML feature that allows on data validation by an XML parser¹³, the text „**body (tool_declaration*, tool_instance+, notification*, task_mask?)**” states that this XML file should contain one root tag that is called **body** which includes following tags: **tool_declaration**, **tool_instance**, **notification**, **task_mask**, more detailed description can be found in next point DTD - 3.1.2.4.

XML as HTML can have exactly one root tag element, but in the case of XML root tag is user definable, according to an example the the root tag name is *body*.

Tag *tool_declaration* is an example of empty tag, the only data stored by a tag are values of *name* and *alias* attributes, please note that attributes are not obligatory, so empty tag without any attributes can be defined too. Empty tag in XML must be terminated with `'/>'` what makes a significant difference in contrast to HTML where empty tag simply is an opening tag.

The comments can be declare in the same way as in HTML, as it is in another languages comments are intended to be readable by human only, so XML analyzing software simply ignores it.

¹³ parsing is a process of analysing a text, the task of XML parser is to analyse the data stored in xml format in order to allow for future data analysis, e.g. to configure the program if xml was used as a configuration file

DTD

Document Type Definition (DTD) is a language to define the structure of an SGML family documents. SGML – Standard Generalized Markup Language is standard for the description of markup language, in other words SGML it's so called metalanguage – the language used to make statements about statements in another language. SGML is a base for Markup Languages such as XML and HTML. Regarding previous DTD definition, it's also true to say that DTD is used to describe the structure XML and HTML documents.

The document content is defined in DTD using declarations, the most important declarations are as follows:

- !DOCTYPE element ... – defines the root element
- !ELEMENT element ... – defines what elements are included inside certain element
- !ATTLIST element ... – defines attributes of a certain element

Inside ELEMENT declarations the quantifiers can be used as it was shown in previous example, the meaning of quantifiers is as follows:

- + – specifies that there must be one or more occurrences of the item
- * – specifies that there must be zero or more occurrences of the item
- ? – specifies that there must be zero or one occurrences of the item
- – if there is no quantifier it specifies that there must be exactly one occurrences of the item

DTD can be declared inside XML document or in a separate file, in the second case the XML file must be supplemented by reference to the complementary DTD file.

3.1.2.5 XSLT

The disadvantage of HTML format is mixing document structure with data, what makes html documents more difficult in maintenance, especially in systems where data are updated often. The answer for this issue is Extensible Stylesheet Language Transformations – XSLT. It's a XML based language used to transform XML documents.

The XML document in conjunction with XSLT is transformed to output document, the output document might be an another XML syntax document, however HTML or plain text formats are also possible.

Harnessing XSLT for transforming XML documents to HTML is particularly useful, as it provides the data and structure separation. In this case XML document contains only data while XSLT defines structure. XSLT provides well known from programming languages loop or selection statements what was shown at the examples bellow:

```
<xsl:if test="count( day[@week=$WEEK_NO]/shift ) != 0">
    <a href="weekly_view.xml#week{day[$day_of_month]/@week}">
        <span class="abs_pos"></span>
    </a>
</xsl:if>

<xsl:for-each select="/calendar/month">
    <a name="month{@no}"></a>
    <table class="cal">
        <tr>
            <th class="cal" colspan="7">
                <xsl:value-of
                    select="document('../year_map.xml')//map-
month[number( current()/@no ) ]"/>,
                <xsl:value-of select="@year"/>
            </th>
        </tr>
        <xsl:call-template name="print_month_table"/>
    </table>
</xsl:for-each>
```

The language is featured by a possibility of calling a certain code, thus an analogy to functions or methods known from structured or object oriented programming languages is a part of XSLT too. The XSLT specification can be found in [W3C - XSLT].

3.1.2.6 AJAX

One of the disadvantages of HTML was a need to reload the whole webpage whenever any data transfer to server was needed. This system was good in the cases where after sending data (e.g. form or questionnaire) only the confirmation page was required. However this approach was not to convenient in the case of more user interactive application where the page reloading every time when small change is committed was disturbing for users, another side effects were an additional traffic and server load.

To address this issue the Asynchronous JavaScript and XML (AJAX) was created. The AJAX is not a technology name, but rather a term to call a set of technologies that allow on data sending from a web applications in a background. This term incorporates the following technologies:

HTML and CSS	– for data presentation purposes
DOM	– as an interface to interaction with data
XML or JSON	– for data interchange, XSLT might be used for data manipulation to support XML
XMLHttpRequest object	– for asynchronous communication, this JavaScript object is a key element of AJAX as it allows on data interchange
JavaScript	– is used to bind these technologies together

As an example of AJAX application the shifts reservation process from *ShiftsScheduler* will be briefly described. Firstly, shifter must find free block of shifts with is respectively marked, after clicking at the marked area, the mouse click event is generated that calls appropriate function with checks what block was selected and forwards this information to server using XMLHttpRequest object. The server checks if a certain block can be reserved by this user and if everything is fine makes an reservation and sends response back to client. Then on the user's side proper JavaScript function is called which using DOM objects marks proper blocks as reserved.

The data can be exchanged in XML format which was already explained, or in JSON descried in a next point. Ajax in the context of Java programming language is described in [Olson, Sven Douglas - AJAX].

JSON

JavaScript Object Notation – JSON, it's a text based standard designed for data interchange, it is based on JavaScript data types. The format is human-readable, for instance the table holding the different kinds of animals is simply represented as the following text:

```
| {“animals”:[“Bird”, “Mammal”, “Fish”, “arachnid”]}
```

Despite JSON association with JavaScript the standard is language-independent, what makes possible to interchange data between JavaScript and a Java server. JSON specification can be found in [Crockford, D. - JSON].

3.1.3 HTTP

The Hypertext Transfer Protocol (HTTP) was developed together with URL and HTML to complement World Wide Web by data transfer capabilities. An HTTP session is an exchange of request and response messages between client and server. The communication is initiated by a client which after establishing TCP¹⁴ connection to port 80¹⁵ sends the request message, the server sends back relevant response.

The request message is characterized by a method which indicates requesting action. The most common method is GET that determines what resource identified by URL the server should return as an response.

The HTTP protocol is so called stateless protocol what means that the server doesn't retain status about connected users. This functionality is not required in the case of services that are oriented strictly on the data sharing, however in the case of a web applications maintaining user's status becomes necessity. This issue can be addressed by a series of technics such as HTTP cookies, URL-rewriting or hidden variables.

As a WWW was getting on popularity the protection of confidential data become necessity. The HTTPS and S-HTTP protocols were introduced to provide encrypted data transmission. HTTPS – Hypertext Transfer Protocol Secure wraps the entire communication within encrypted connection. S_HTTP – Secure Hypertext Transfer Protocol in contrast encrypts only content of the message and some fields. HTTPS is nowadays dominant protocol leaving S-HTTP marginalized.

3.2 Java – application tier

Java was released by Sun Microsystems in 1995 as so called software platform that consist of virtual machine, class libraries and most importantly the Java language. The objectives of Java was to develop familiar, secure and platform independent object oriented language. Java is an object oriented¹⁶ language, as the syntax is derivative of C/C++ it makes it very familiar. In contrary to C/C++ Java is not compiled to native code that is directly executed by microprocessor, but to so called bytecode which is interpreted by Java Virtual Machine. Java Virtual Machine ensures security and provides platform independence.

Java is very widely used in many applications such as business software or embedded systems. As Java was designed with a focus on security it's a very suitable platform for a web based applications. *ShiftsScheduler* and *ElogLooker* application uses Java Servlet technology, servlets are Java components installed at the server side that generate the response to HTTP requests. The basics of Java can be found in [Niemeyer, Patrick; Knudsen, Jonathan - Java], the Java Servlets are described with details in [Hunter, Jason; Crawford, William - Java Servlet].

14 TCP - Transmission Control Protocol is one of the essential protocols of TCP/IP model that specifies general guidelines for a different purpose networking protocols. The set of TCP/IP protocols enables computer – computer connectivity over network

15 port 80 of TCP protocol is reserved for HTTP communication, ports in TCP protocol are used to identify sending and receiving application end points

16 in fact Java is not fully object oriented language as some basic data types like integers, floating point numbers or boolean values due to performance reasons are not classes

Frameworks

When developer have to focus on a certain task the common approach is use the libraries where details that are not relevant to project can be hid. The framework is a mechanism which like the library is used as an abstraction layer to relieve developers work, but unlike the library the framework provides:

- the flow control can be redirected to framework – it means that the software developer wifes the full control over the program as the framework may need to perform some tasks that the developer shouldn't brother about
- the framework has a default behavior, so even if developer will not pass a parameters to framework the default action will be automatically performed
- the framework can be extended by user in order to adopt it to specific needs
- the framework code can't (or it's strongly not advised) be modified as only framework's developers who knows it very well can apply safe modifications, this the reason why framework should allow on extinction mentioned in a previous point

The *ShiftsScheduler* and *ElogLooker* uses two frameworks, first the Apache Cocoon to simplify the web application development and the Hibernate to make the database access more easy.

3.2.1 Apache Cocoon Framework

The Apache Cocoon framework is build around the components separation and pipelines, it's focused on XML and XSLT publishing.

The Cocoon's functions are realised by specialized components, the components are interacting with each other in order to provide the response for a certain http request. To bind components together the pipeline is used. The most important Cocoon's components are:

Matchers

Matchers are used to match the http requests against wildcard or regular expressions.

Generators

Generators creates the data stream for future processing, for instance it can be the XML stream that can be intended for XSLT processing by a next component (see Transformers point).

Transformers

Transformers are used to transform a stream of data to another format, e.g. from XML to HTML using XSLT processing.

Serialises

The Serializer is the end point for an XML stream, it prepares data for final client consumption.

Typically a pipeline consists of a generator, then zero or more transformers, and eventually the pipeline is closed by serializer.

3.3 Databases – storage tier

The database according to [McGraw-Hill - Directory] is:

“A nonredundant collection of interrelated data items that can be shared and used by several different subsystems.”

The database is very convenient solution for application developers as they doesn't need to deal with such an issues as data seeking or storage. All this basic tasks are handled by the database system while application's developer can focus on tasks directly connected to developing by him software that are: defining data structure, manipulating data, data protection rules.

These goals can be achieved with the data manipulation language, the most popular language of this family is SQL – Structured Query Language which is commonly used with relational database systems.

In relational databases that are dominant the data are organised in the form of tables, the columns represents the attributes while data are stored in a rows. The table from Illustration 10 on page 36 can be a simple example of a database table¹⁷, the table header represents an attribute names associated with a certain column, while the data are stored in rows. The tables can be related each other, thus more complex data structures can be created.

At this point It's worth to recommend one of the most comprehensive books on database system implementation written by three well known computer scientists from Stanford University: [Gracia-Molina, Hector; Ullman, Jeffrey D.; Widom, Jennifer - Database].

3.3.1 PostgreSQL

The PostgreSQL, (to be pronounced as /'pɒstɡrɛs ˌkjuː 'ɛl/) is one of the leading database systems available nowadays. An advanced capabilities makes it suitable for a professional use, the software is issued under open source licence what is an additional benefit.

The ShiftsScheduler and ElogLooker applications uses the following capabilities of the PostgreSQL:

- **VIEW**, view can be considered as a virtual database table, it can be used to join two independent tables, thus it can lead to simplify the application's code and shorten overall development time. Another example of views is showing a selective table content, so unwanted informations can be simply hidden. View consist of a stored SQL query (that can be used for tables joining or content filtering), that is executed when user or application is trying to view view's content.

Views can be treated like the normal table, however in contrast to table it's not possible to insert or remove rows. Although this limitation can be overcome with usage of triggers or rules (see point TRIGGER and RULE).
- **FUNCTION**, like in structured programming languages such as Pascal or C, the SQL functions are used to perform a certain operations, in the case of database this are operations over the data set. The functions are usually called by the application to perform more complex operations like inserting data to a set of different tables, the function can also calculate or find values which should fill the table.

¹⁷ please notice that it's a very simple example, the ShiftsScheduler database system is much more complex

- **TRIGGER**, the trigger is a peculiar example of function, it's executed only when a certain event occurs such as a row update or row insertion, so the database can automatically perform some necessary operations on data. The triggers are very useful for data validating, e.g if the database is intended to hold an e-mail addresses, the trigger can be used to validate if the text that was given represents the correct e-mail address.
- **RULE**, rule can be used to redefines the database behaviour for a certain actions such as data inserting or deleting. For instance it's possible to define rule that is doing nothing when attempted to remove a row occurs, this prevents the database system from performing removal operation.

Presented database features allows on a good separation of a data structure and application. Although another database systems implements SQL standard as well, there is a slight difference between a various implementations.

3.4 Eclipse – development environment

The developers productivity depends on his/her skills, experience and motivation. The good development tools are playing an important role too.

For my work I use Eclipse development environment, it's an integrated development environment (IDE), it's characterized by an extendible plug-in system. It can be used to develop applications in Java and various different programming languages such as C, C++, PHP, Ruby. Eclipse began as IBM's project initiated in 2001, in 2004 the Eclipse Foundation was created to take it over. The Eclipse is a free software and open source software.

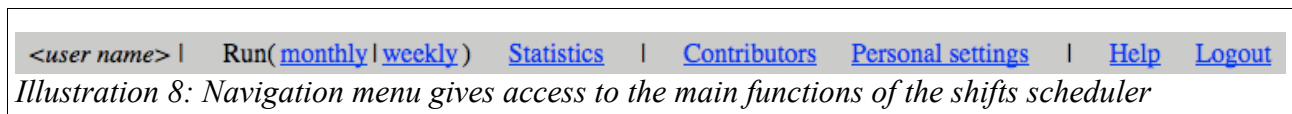
4 User's interface

This chapter describes the *ShiftsScheduler* and *ElogLooker* user's interface, it can be treated as guide for an applications users.

4.1 Shifts Scheduler

This point explains how to use the *ShiftsScheduler* web service, please note that such a details like how to login were intentionally skipped.

The next subchapters are the reflection of the navigation menu which provides access to all functionalities of the Shifts Scheduler, this menu is located at the top of the web page (see Illustration 8).



In order to login first time the temporary password has to be used, it's highly recommend to change a password to a new stronger one according to CERN passwords recommendations (<https://security.web.cern.ch/security/recommendations/en/passwords.shtml>).

Please notice that for the Shifts Scheduler as for another web services due to the security reasons it's highly recommended to **logout** when work with service is finished.

The functions of the Shifts Scheduler are as follows:

run	–	allows on shifts scheduling
statistics	–	shows the shifts statistics
contributors	–	shows the list of institutes and team members
personal settings	–	allows users to change the account settings
help	–	shows this help
logout	–	allows to logout from the service

4.1.1 Run option

The run option allows users to reserve and confirm the shifts as well as to have an overview of the schedule. This is the crucial function of the *ShiftsScheduler*, all another ones are just an complementary, so it's important to be introduced to this issue.

The run is represented in the calendar form, depending on the current view you can see the calendar in the **monthly** or **weekly** mode. The mode can be changed by choosing „monthly” or „weekly” in the navigation menu. Another way to change the mode is by clicking small magnifying glass that is in the calendar near the information about week number.

The monthly mode allows on having an overview of a free and reserves dates, while the weekly view gives an additional detailed informations about who and from what institute is signed up for a certain shift. The monthly view is convenient to quickly check the shifts occupancy, while the weekly view works better when more details is needed. Please note that by pointing the mouse indicator over the cell representing shift the hint appears on the left side.

Colour codes and symbols

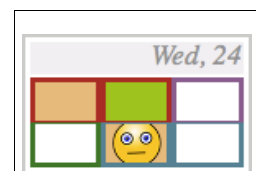
To describe the shifts status in a user friendly way the colour codes were introduced. The meanings of these colours are as follows:

white	–	free (the place is free and it's ready to be taken)
 light brown	–	reserved
 green	–	confirmed

On the top of the colour marks the face iconography symbols are used, the icons are used to represent the shifts of logged person.

	unsure face	indicates that the shift was reserved by you or for you as this icon points your reservation, it always appears at the light brown background
	smiley face	indicates that the shift which you reserved or was reserved for you has been confirmed as this icon points confirmation for you, it always appears at the green background

The Illustration 9 shows an example reservation. By following the example, the first place for a night shift is reserved by somebody while the second one is free. For a day shifts the first place is confirmed for somebody while the second one is reserved for the shifter who is logged in, what is indicated by symbol . The night shifts are not reserved at all.



*Illustration 9:
Example of a
shifts and
shifters
representation in
a Run calendar
for a monthly
view*

Reservations and cancellations

The free places (indicated by a white colour) can be reserved by clicking the 'R' (for a monthly view) or 'Reserve' (for a weekly view) link. This action triggers immediate reservation what is presented by a colour change from white to light brown and in a “unsure face” icon appearance. In addition information about the reservation appears in the hint menu at the left side.

The hint menu gives the possibility to cancel the reservation by clicking [cancel]. Note that all reservations are the block reservations, the 'R' or 'Reserve' appears in the cell that represents first shift only. So if block covers three shifts the reserve button is not repeated three times but once at the beginning only.

Confirmations and rejections

The block confirmation or rejection can be done from authorized account (group leader or administrator, see Table 2 in chapter 2.1.3) only in the weekly view mode.

When authorized user is in a weekly mode the blocks that can be confirmed or rejected contains two links: “confirm this reservations block” and “reject this reservations block”.

When one of those links is clicked the system asks about confirmation, if an answer is positive a proper action is performed by a system. As in the case of reservations the confirmations and rejections links appear in the first shift of the block only.

Actions on “behalf of” other shifters

The reservations on behalf of another person can be make from authorized account (group leader or administrator, see Table 2 in chapter 2.1.3). When authorized person is logged in, an additional option „set in charge of ...” in the top menu is visible. This option allows to select shifter in charge of whom actions in Shifts Scheduler will be performed.

When account is set to perform the actions in behalf of other shifter set of a small changes appears in a service to indicate that now any action is performed for another user. In a run calender in monthly view the reservation link appears not as 'R' but as 'R !', in a weekly view link 'Reserve' is replaced by 'Reserve for <shifter name>'. The option 'Personal informations' (see chapter 4.1.4) from the top menu is replaced by '<shifter login name> settings'. The difference is that the option '<shifter login name> settings' in contrast to 'Personal informations' not allows on data editing, so only the information about “behalf of” shifter profile can be checked.

To quit from “behalf of” mode user can either logout, then after the next login this mode will be off, or just click “your credentials” in the top menu from left side.

4.1.2 Statistics

This function allows on having a view at the statistics for a current year or separately for selected shifts. The statistics shows how many shifts have been reserved and confirmed by institutes as well as the amount of points gained and required. An example statistics from Shifts Scheduler is presented at the Illustration 10.

institute	taken shifts:			points:	
	reserved (nights only)	confirmed (nights only)	sum (nights only)	obtained	required
Belgrade, Serbia - University of Belgrade	20(5)	0(0)	20(5)	17.5	0
Bergen, Norway - University of Bergen	9(3)	0(0)	9(3)	9	6
Budapest, Hungary - Konkoly Thege Miklos	24(14)	0(0)	24(14)	26	36
Dubna, Russia - Joint Institute for Nuclear Research	15(9)	0(0)	15(9)	15	54
Frankfurt am Main, Germany - Fachhochschule Frankfurt am Main	1(0)	9(3)	10(3)	9	6
■ ■ ■					
Warsaw SINs, Poland - Soltan Institute for Nuclear Studies	14(4)	6(0)	20(4)	19	18
Warsaw UW, Poland - University of Warsaw	26(7)	6(0)	32(7)	30.5	42
Warsaw WUT, Poland - Warsaw University of Technology	62(10)	17(0)	79(10)	79.5	54
Wroclaw, Poland - University of Wroclaw	6(3)	0(0)	6(3)	5.5	0
Zagreb, Croatia - Rudjer Boskovic Institute	17(6)	0(0)	17(6)	18	18
Zurich, Switzerland - ETHZ Hoenggerberg	6(0)	0(0)	6(0)	5	36
total:	436(166)	160(34)	596(200)		

Illustration 10: Statistics table (the image is intentionally cut)

The meaning of the table columns presented at the Illustration 10 is explained below.

institute	– the name and location of the institute
shifts: reserved (nights only)	– amount of shifts reserved by a certain institute, in the brackets the amount of reserved night shifts only is presented
shifts: confirmed (nights only)	– amount of confirmed shifts for a certain institute, in the brackets the amount of confirmed night shifts only is presented
shifts: res. & conf. (nights only)	– amount of reserved and confirmed shifts for a certain institute, in the brackets the amount of reserved and confirmed shifts is presented only
points: obtained	– amount of points that institute gained or will gain if the members will pass declared shifts
points: required	– the amount of points that is required to gain by a certain institute

At the bottom of the statistics table the total sum of all reserved and confirmed shifts is presented.

4.1.3 Contributors

Under this option the list of contributing to NA61 institutes as well as a list of it's shifting members is presented. The list indicates who is a group leader as well as ShiftsScheduler administrator.

4.1.4 Personal informations

Personal settings option allows on editing personal data, it gives a possibility to edit a first and second name, an e-mail address and a mobile phone number. It gives also a chance to change a password.

4.2 Elog Looker

The *ElogLooker* is a small application designed for simple data statistics presentation. The generic design makes the application easy in adaptation to new environment. Now, the program is in an experimental stage what is exposed by lack of parameter filters described later.

The program consist of one table, the header and a first row shows grouped values of selected properties. At the intersection of a row and a column the statistic for a certain values of parameters are presented. The statistic is a simple sum and count (value in a round brackets) of the third parameter.

This description might looks complex, to help to clarify it an example application print screen is shown at Illustration 11. The rows are label by a values of a Beam Energy that are: 10, 20, 31, ..., 350. The columns are described by a Beam Type, possible beam types are: 'No beam', h, h+, h-, p, pi+, pi-. The cells of a column presents the sum of a of value “Number of Events” for a certain beam energy and a certain beam type. Please note that the “Number of Events” is a number, so the summary operation is possible. Another statistic that is presented in a round brackets is a count, the count here means the number of occurrences of a certain value for e certain beam energy and a type, for this operation element can be of any type because the only factor is how many times the certain value was counted.

Summary of NA61 elog reports for Physics Run for Run protocols								help & about:
rows:	Stop Reason Beam Type Beam Energy Target Type Target Thickness Trigger Type							
columns:	Stop Reason Beam Type Target Type Target Thickness Trigger Type							
statistics from:	Run Number Run Type Number of Events							
[GeV/c]:	No beam	h	h+	h-	p	pi+	pi-	
10			168,470 (6)					
20		360,457 (8)			2,434,537 (141)			
31	108,335 (37)				16,940,176 (617)	1,135 (3)		
40					5,846,468 (114)			
75			460,307 (106)					
80					5,002,703 (105)			
158				3,314,502 (44)	3,977,859 (79)		2,722,902 (44)	
350				4,978,554 (62)			96,938 (2)	

Illustration 11: ElogLooker application view

Attributes filters

The application was designed make a statistics from attributes that are stored in database, the program is considering all attribute configurations without verifying if it makes sense from a “human” point of view. As an example the Illustration 12 can be brought up where the “Run number” can be selected to make a statistics, although it doesn't makes sense from users point of view to analyse this value.

Summary of NA61 [elog](#) reports for Physics Run for Run protocols [help & about](#)

rows:	Stop Reason Beam Type Beam Energy Target Type Target Thickness Trigger Type
columns:	Stop Reason Beam Type Target Type Target Thickness Trigger Type
statistics from:	Run Number Run Type Number of Events

[GeV/c]:	No beam	h	h+	h-	p	pi+	pi-
10			60,171 (5)				
20		73,316 (8)			1,310,802 (141)		
31	314,665 (37)				5,268,939 (617)	21,407 (3)	
40					1,090,678 (114)		
75			769,248 (106)				
80					1,058,686 (105)		
158				434,585 (44)	786,009 (79)		432,492 (44)
350				605,189 (52)			19,411 (2)

Illustration 12: ElogLooker application, the statistics of run number are pointless from man's point of view, the attributes filters can be used to fix this issue

The solution is to introduce the Attributes filters that can be set by an administrator to hide miss-leading functionality.

5 Conclusions

The World Wide Web since its release 20 years ago evolved from a simple internet data sharing system to an expanded internet application platform that revolutionized the information sharing and communication. The invention of a WWW was a base for founding such companies as Google, Yahoo! or eBay which boosted the „internet revolution”. Nowadays the Web is undoubtedly one of the most popular internet services, the web browsers or web widgets¹⁸ are now a standard software of devices other than a PC computers such as a mobile phones or even TV's.

When it comes to decide what technology should be used to develop the shifts scheduling and *Elog* data viewing application, the decision to use WWW was unquestionable. The development of a web applications is special as it is a compilation of a different languages and standards where each of those is designed to perform specialized tasks. Unlike the standard application written in languages such as C++ or Java, where developer deals with the same technology regardless if it's an application's logic or user's interface layer.

My experience with JavaScript and AJAX as well as with advanced frameworks such as Cocoon or Hibernate improved my skills and knowledge about this very important field which is WWW. The experiences with advanced functions of database such as triggers are another values of my master thesis work.

And the last but not least, the fact of working on an applications that will be implemented to serve to its users is very motivating. The communication is an essential in a team work, presenting an ideas, consulting and verifying them against point of view of other group members who are representing a very different fields than the Computer Science are a very important skills in professional work of software developer.

¹⁸ web widget is a small application developed using web technologies that is designed to perform simple work, for instance to deliver the content such as the weather or news

Bibliography

- Berners-Lee, T.; Masinter, L.; McCahill, M. - URL, „Uniform Resource Locators”,
<http://www.ietf.org/rfc/rfc1738.txt>, December 1994
- CERN - Knowledge, „CERN Knowledge Transfer”, <http://knowledgetransfer.web.cern.ch/>,
[accessed on 2011/12/02]
- Champeon, Steve - JavaScript, „JavaScript: How Did We Get Here?”,
http://oreilly.com/pub/a/javascript/2001/04/06/js_history.html, April 2001
- Crockford, D. - JSON, „The application/json Media Type for JavaScript Object Notation (JSON)”,
<http://tools.ietf.org/html/rfc4627>, July 2006
- Ecma - ECMAScript, „ECMAScript Language Specification, Edition 5.1”, <http://www.ecma-international.org/publications/standards/Ecma-262.htm>, June 2011
- Flanagan, David - JavaScript, „JavaScript: The Definitive Guide, 4th Edition”, O'Reilly Media,
ISBN 978-0-596-00048-6, 2001
- Gracia-Molina, Hector; Ullman, Jeffrey D.; Widom, Jennifer - Database, „Database systems: The complete book, 2/E”, Prentice Hall, ISBN 0131873253, 2008
- Hunter, Jason; Crawford, William - Java Servlet, „Java Servlet Programming 2nd Edition”, O'Reilly & Associates, ISBN 0596000405, 2001
- McGraw-Hill - Directory, „Directory of Scientific and Technical Terms, Fifth Edition”, Parker, Sybil P. - Editor in Chief, McGraw-Hill, Inc, ISBN 0-07-113584-7, 1994
- Niemeyer, Patrick; Knudsen, Jonathan - Java, „Learning Java, Third Edition edition”, O'Reilly Media, ISBN 0596008732, 2005
- Olson, Sven Douglas - AJAX, „Ajax on Java”, O'Reilly Media, Inc., ISBN 978-83-246-1110-2, 2007
- Piwek, Mateusz - kontrola komputerów, „System zdalnej kontroli komputerów w sieci DCDB dla eksperymentu ALICE/eng. Remote computer control system for DCDB networks” - Bachelor Thesis, Warsaw University of Technology - Faculty of Mathematics and Information Science, 2007
- PSI - ELOG administration, „ELOG Administrator's Guide”,
<https://midas.psi.ch/elog/adminguide.html>, [accessed on 2011/11/12]
- PSI - ELOG usage, „ELOG User's Guide”, <https://midas.psi.ch/elog/userguide.html>, [accessed on 2011/11/12]
- W3C - CSS, „Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification”,
<http://www.w3.org/TR/CSS2/>, 07 June 2011
- W3C - HTML, „HTML Tag syntax”, http://www.w3.org/wiki/HTML/Training/Tag_syntax,
[accessed on 2011/11/22]
- W3C - XSLT, „XSL Transformations (XSLT) Version 1.0”, <http://www.w3.org/TR/1999/REC-xslt-19991116>, 16 November 1999
- W3C; XML, „Extensible Markup Language (XML) 1.0 (Fifth Edition)”,
<http://www.w3.org/TR/2008/REC-xml-20081126/>, 26 November 2008