

Web System for Data Quality Assessment of Tile Calorimeter During the ATLAS Operation

C. Maidantchik¹, F. Ferreira¹, F. Grael¹, A. Sivolella² & L. Balabram²
on Behalf of ATLAS TILE Calorimeter Community

¹COPPE, Federal University of Rio de Janeiro (UFRJ), Brazil

² Escola Politécnica, Federal University of Rio de Janeiro (UFRJ), Brazil

E-mail: Tile.Commissioning@cern.ch

Abstract. TileCal is the barrel hadronic calorimeter of the ATLAS experiment and has about 10 000 electronic channels. Supervising the detector behavior is a very important task to ensure proper operation. Collaborators perform analysis over reconstructed data of calibration runs in order to give detailed considerations about failures and to assert the equipment status.

Since the commissioning period, our group has developed seven web systems that guide the collaborators through the data quality assessment task. Each system covers a part of the job, providing information on the latest runs, displaying status from the automatic monitoring framework, giving details about power supplies operation, presenting the generated plots and storing the validation outcomes, assisting to write logbook entries, creating and submitting the bad channels list to the conditions database and publishing the equipment performance history.

Calibration runs are more often during the ATLAS operation. This represents a substantial increase amount of data, hence a challenge for the systems. An optimized data model was designed to reduce the number of needed queries. The web systems were reassembled in a unique system to provide an integrated view of the validating process. The server load was minimized by using asynchronous requests from the browser.

1. Introduction

ATLAS [1] is one of the particle physics experiments at the Large Hadron Collider at CERN, designed to measure proton-proton and heavy ions collisions¹. ATLAS is composed by several detectors with particular features.

Tile Calorimeter [2] (TileCal) is the central section of the hadronic calorimeter. It is a sampling calorimeter using iron as absorber and scintillating tiles as active medium, that is readout by wavelength shifting fibers and photomultiplier's [3]. TileCal is composed of four sections divided in two Long Barrels (central section) and two Extended Barrels. Each barrel is divided azimuthally into 64 wedges, so-called modules. A module contains a number of cells where each cell has two PMTs. Hence, there are roughly 10 000 readout channels in the calorimeter.

The number of electronic channels and the complexity of the experiment demand the continuous monitoring of the acquired data's quality. Collaborators perform analyzes over reconstructed data of calibration runs in order to give detailed considerations about failures and to assert the equipment status.

¹ In 2010, the proton-proton collisions reached 7 TeV center of mass energy.



For achieving this goal, hundreds of plots and histograms are analyzed, the output from Data Quality Monitoring (DQM) Framework is considered and complete reports are composed. Findings are presented at the weekly meetings and discussed among the TileCal community . Then, the data quality responsible provides the list of problematic channels that should not be considered for the future physics analysis.

During the commissioning period, seven web systems were developed that guide the collaborators through the laborious data quality assessment task. The event rate during the ATLAS operation increased amount of data produced. Over ninety million plots are stored in the databases, affecting the performance of the systems. Thus, an optimized data model was designed to reduce the number of needed queries. The web systems were reassembled in a unique system to provide an integrated view of the validating process.

Hence, a unique web system was created. It presents a set of views, each one representing a different aspect of the duty. For example, one view displays information about the latest runs and whether they were reconstructed. Other one shows the available plots and histograms and allows the user to register comments. The system will be presented by detailing its architecture(Section 2) , its data model (Section 3) and the technologies that were used for its implementation (Section 5). The different views of the system are expositied in the Section 4.

2. Software Architecture

The developed system is based on the Model-View-Control (MVC) software architecture and uses a javascript framework for rendering the user interface and performing requests to the server asynchronously. It uses a python based back-end for responding these requests.

The MVC pattern isolates the application logic from the user interface, allowing them to be developed, tested and maintained separately. The *model* represents the system's data structure and how the application will access it. It manages the behavior from the requests of the end user. The *view* renders the models making it suitable for the user's interaction. The user's actions are processed and requests are sent to the model whenever it's necessary. The *controller* is the set of scripts that activates the model according to the requests from the client. Figure 1 describes how the different layers communicate among themselves.

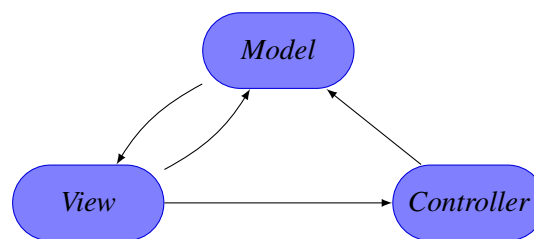


Figure 1. Model-View-Controller Architecture

The proper functioning of the system depends on how the data is modeled. Therefore, after the beginning of the ATLAS operation, the redesign of the database schema was done very carefully. It was intended, among other things, to minimize the server load and to reduce the number of records. This improved data model is described in the next section.

3. Data Model

The web system uses a relational database for data persistence. The model presents two major entities – run and module – which are associated by a many-to-many relationship. One run has the following attributes: date, type, id number and the number of events. For guaranteeing the normalization, the entity *run type* was created since they are limited. The module is represented by the name of the barrel where it is located and by its drawer number, e.g., LBA25. The Figure 2 shows the relations.

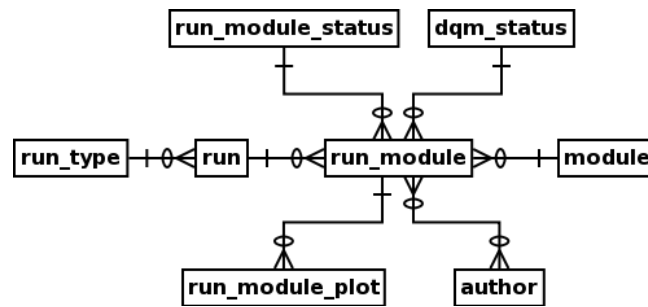


Figure 2. Data model after the beginning of ATLAS operation.

After the run reconstruction, hundreds of plots and histograms are created and stored into public directories in a local server at CERN. The graphics' location are stored in the database. Also, the DQM Framework [2] generates flags for several different test in a channel level. For each module, a summary flag is set according to the channels' tests.

The data quality validator attributes status for each module in a run according to its performance after analyzing the graphics and understanding the DQM flag. There is five possible values for this variable: "OK", "Bad", "Some Problems", "Not Analyzed" and "Not to be Analyzed". The shifter is also able to provide detailed comments if needed. The shifters' names and their CERN ids are stored.

Thus, the new data schema represents the entities from the data validation task. The software's model pattern uses this schema and takes care how the attribute values are filled and how the system responds for different requests. The next section presents the different views where the user perform actions. These actions will be processed by the view engine and, if necessary, requests will be sent to the model.

4. Web System Views

This system aims to be a guideline for the data validator during the offline shift. Each view deals with different stages of the task. This way, the only thing needed to complete the validation job is to go through the different screens for a given run. First, the shifter checks whether there is a new run and, moreover, whether it has been reconstructed. The system presents the latest three hundred runs in a tabular format, as it can be seen in the Figure 3.

The table presents the most relevant details about the run: the run type, the run date and the number of events. Under the *Plots* column, a hyperlink is made available whenever the respective run is reconstructed. That's the way validators know when the plots and histograms are done. Also, through this link they can access the plots and perform their analysis, as it will be explained later in this section.

The systems shows an overview for each analyzed runs by calculating the proportion of good and bad modules according to the statuses foreseen in the software model. It also displays the name of the responsible of the analysis. It is important to notice that a run can have more than one validator. This way, the whole collaboration is able to follow the DQ team finds about the TileCal equipments.

It's also possible to follow the operation of the detector over the time, through another perspective. The user can access a timeline display by clicking on the link available in the top page of the first screen. A search interface is presented where is possible to select the models, the run type and the time period of interest.

After performing the search, the system displays the performance of the modules throughout the consecutive runs for the chosen time period. The statuses given by the validator are displayed according the following color code: *OK* is green, *Bad* is red, *Some Problems* is yellow, *Not to Be Analyzed* is black and the *Not Analyzed* is white. Runs with detailed comments are marked by a "pencil" icon. The Figure 4 shows the Timeline view.

Back to the view shown in the Figure 3, by clicking in the icon under the *Plot* column, it is possible to access the view where a run can be analyze. This view presents all modules for a given run, as it can

Run Number	Run Type	Date	# of Events	Overview						Plots	Shifter
				OK	Some Problems	Bad	Not to be Analyzed	Not Analyzed	# of Modules		
169594	Las	2010-11-21 14:43:34	38952	0.00%	0.00%	0.00%	0.00%	100.00%	256		
169592	Las	2010-11-21 14:34:25	10111	0.00%	0.00%	0.00%	0.00%	100.00%	256		
169590	Las	2010-11-21 14:30:21	10102	0.00%	0.00%	0.00%	0.00%	100.00%	256		
169588	CIS	2010-11-21 14:25:58	4932	0.00%	0.00%	0.00%	0.00%	100.00%	256		
169584	MonoCIS	2010-11-21 14:18:22	10204	0.00%	0.00%	0.00%	0.00%	100.00%	256		
169580	MonoCIS	2010-11-21 14:09:43	10076	0.00%	0.00%	0.00%	0.00%	100.00%	256		
169578	Ped	2010-11-21 14:02:27	10030	0.00%	0.00%	0.00%	0.00%	100.00%	256		
169555	Ped	2010-11-20 12:52:10	10082	76.95%	19.99%	0.00%	3.52%	0.00%	256		David Anthony Milstead
169546	CIS	2010-11-20 12:26:29	4932	87.11%	9.38%	0.00%	3.52%	0.00%	256		David Anthony Milstead
169543	MonoCIS	2010-11-20 12:23:51	10239	0.00%	0.00%	0.00%	0.00%	100.00%	256		
169538	Las	2010-11-20 12:12:30	10017	0.00%	0.00%	0.00%	0.00%	100.00%	256		
169536	Las	2010-11-20 12:08:31	10011	0.00%	0.00%	0.00%	0.00%	100.00%	256		
169534	Las	2010-11-20 11:44:33	100056	0.00%	0.00%	0.00%	0.00%	100.00%	256		
169521	Las	2010-11-20 11:13:30	58428	0.00%	0.00%	0.00%	0.00%	100.00%	256		
169517	Las	2010-11-20 10:44:40	10067	0.00%	0.00%	0.00%	0.00%	100.00%	256		
169487	Las	2010-11-19 12:47:22	10009	0.00%	0.00%	0.00%	0.00%	100.00%	256		
169486	Las	2010-11-19 12:43:14	10057	0.00%	0.00%	0.00%	0.00%	100.00%	256		

Figure 3. List of the latest runs. The run details, like date, number of events and date are shown.

Run Number	Run Date	Run Type	Run Type	Run Type	Run Type	Run Type	Run Type	Run Type	Run Type	Run Type	Run Type	Run Type	Run Type	Run Type	Run Type	Run Type	Run Type	Run Type
169625	2010-11-21 18:31:21	Las																
169624	2010-11-21 18:27:36	Las																
169623	2010-11-21 18:24:23	MonoCIS																
169594	2010-11-21 14:43:34	Las																
169592	2010-11-21 14:34:25	Las																
169590	2010-11-21 14:30:21	Las																
169588	2010-11-21 14:25:58	CIS																
169584	2010-11-21 14:18:22	MonoCIS																
169580	2010-11-21 14:09:43	MonoCIS																
169578	2010-11-21 14:02:27	Ped																
169555	2010-11-20 12:52:10	Ped																
169546	2010-11-20 12:26:29	CIS																
169543	2010-11-20 12:23:51	MonoCIS																
169538	2010-11-20 12:12:30	Las																
169536	2010-11-20 12:08:31	Las																
169534	2010-11-20 11:44:33	Las																
169521	2010-11-20 11:13:30	Las																
169517	2010-11-20 10:44:40	Las																
169487	2010-11-19 12:47:22	Las																
169486	2010-11-19 12:43:14	Las																

Figure 4. Timeline View.

be seen in the Figure 5.

Data is organized in a tabular format with nine columns: *Run Number*, *Module*, *Run Date*, *Run Type*, *Plots*, *Detailed Comments*, *Status Comment*, *DQM Status*, *Shifter Name*. The first five columns shows data that are automatically filled into the database, while the other ones present data from previous analysis. The user can toggle any column that is not needed. The columns *Status Comment* and *DQM Status* display icons with the same color code presented in the Timeline view. The icons under the *Status Comment* column are a bigger than the *DQM Status* one, because they must be considered more relevant for the collaboration.

This view is integrated to the Timeline one. It is possible to check the Status and DQM statuses of the last five runs with the same type, e.g. Pedestal, for a single module by pressing the “+”.

It is also possible to filter rows by the module's name by writing in the top-left part of the page. For example, if the user writes "LB", just the rows corresponding to the long barrel modules. If an "A" is added to the query, just modules in the A side will appear. The number of rows in the same page can be set as well.

Run Number	Module	Date	Run Type	Plots	Detailed Comment	Status Comment	DQM Status	Shifter
169555	LBC39	2010-11-20 12:52:10	Ped	Select Plot	None	Green	Red	David Anthony Milstead
169555	LBC38	2010-11-20 12:52:10	Ped	Select Plot	high gain noise 47>2.5	Yellow	Red	David Anthony Milstead
169555	LBC37	2010-11-20 12:52:10	Ped	Select Plot	None	Green	Green	David Anthony Milstead
169555	LBC36	2010-11-20 12:52:10	Ped	Select Plot	None	Green	Green	David Anthony Milstead
169555	LBC35	2010-11-20 12:52:10	Ped	Select Plot	high gain noise ch>42 >2.5	Yellow	Red	David Anthony Milstead
169555	LBC34	2010-11-20 12:52:10	Ped	Select Plot	high gain noise ch 46 >2.5	Yellow	Red	David Anthony Milstead
169555	LBC33	2010-11-20 12:52:10	Ped	Select Plot	None	Green	Green	David Anthony Milstead

Figure 5. View for analyzing a given run. Screen for an anonymous user.

The user must log the system in for being able to actually perform analysis. CERN Nice's user name and password are enough for it. The log-in form is reached by clicking the button "Start Analysis" button in the top-right corner of the page. After being authenticated, a new set of functions is made available. Figure 6 shows the analysis view for an authenticated user.

Run Number	Module	Date	Run Type	Plots	Detailed Comment	Status Comment	DQM Status	Shifter
169555	LBC55	2010-11-20 12:52:10	Ped	Select Plot	Ch 20 - low gain rms ~1.8	Green, Yellow, Red, Pencil	Red	David Anthony Milstead
169555	LBC54	2010-11-20 12:52:10	Ped	Select Plot	Ch 26 - low gain rms ~1.8	Green, Yellow, Red, Pencil	Red	David Anthony Milstead
169555	LBC53	2010-11-20 12:52:10	Ped	Select Plot	Ch 34 - high gain rms ~2.5	Green, Yellow, Red, Pencil	Red	David Anthony Milstead
169555	LBC52	2010-11-20 12:52:10	Ped	Select Plot	Ch 36-38 - low gain amp ~300	Green, Yellow, Red, Pencil	Red	David Anthony Milstead
169555	LBC47	2010-11-20 12:52:10	Ped	Select Plot	None	Green, Yellow, Red, Pencil	Red	David Anthony Milstead
169555	LBC46	2010-11-20 12:52:10	Ped	Select Plot	Ch 4,46 - low gain amp ~300	Green, Yellow, Red, Pencil	Red	David Anthony Milstead

Figure 6. View for analyzing a given run. Screen for an authenticated user.

Two major differences comparing to the previous screen: the top toolbar and the "pencil" icon for the cells under the "Detailed Comment" and the "Status Comments" columns. This symbol shows that the

user can add/edit information for those specific fields. The Shifter's name is automatically filled after the end of the run analysis. For adding/editing a "Detailed Comment", it is just needed to click over the cell and a hidden text-area field appears. For selecting/changing the status comment value is just needed one click over the small icons.

Plots and histograms are available at the select boxes under the "Plot" header. By selecting one plot a new view is shown. This view shows the selected plot in its real size and the other available plots as a tiny icon, so the user can choose any graphic without change from this screen. It is also possible to open all plots at a once as some users prefer. The Figure 7 shows this view in details.

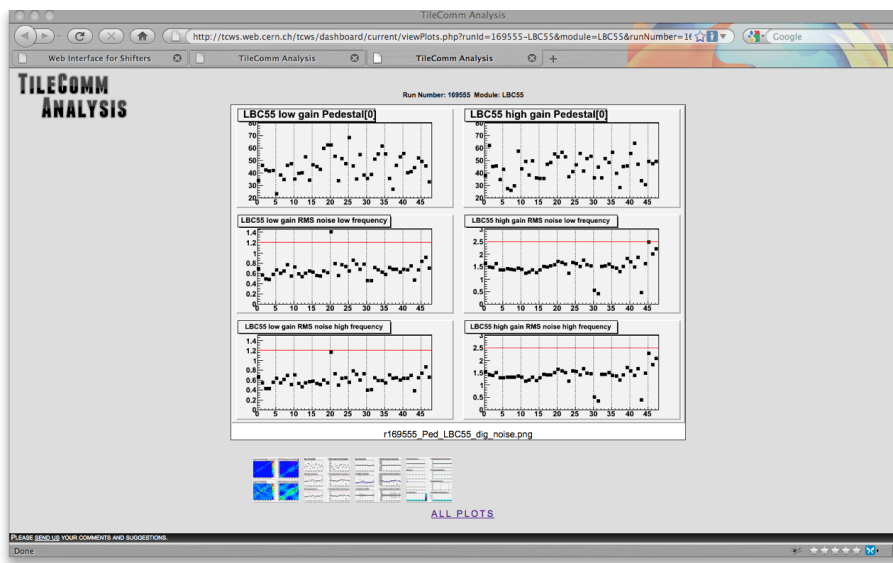


Figure 7. View for displaying plots.

The top toolbar presents four functionalities, each one represented by a different button. The first one summarizes all the shifter's notes and post them to the last view of the process, shown in the Figure 8.

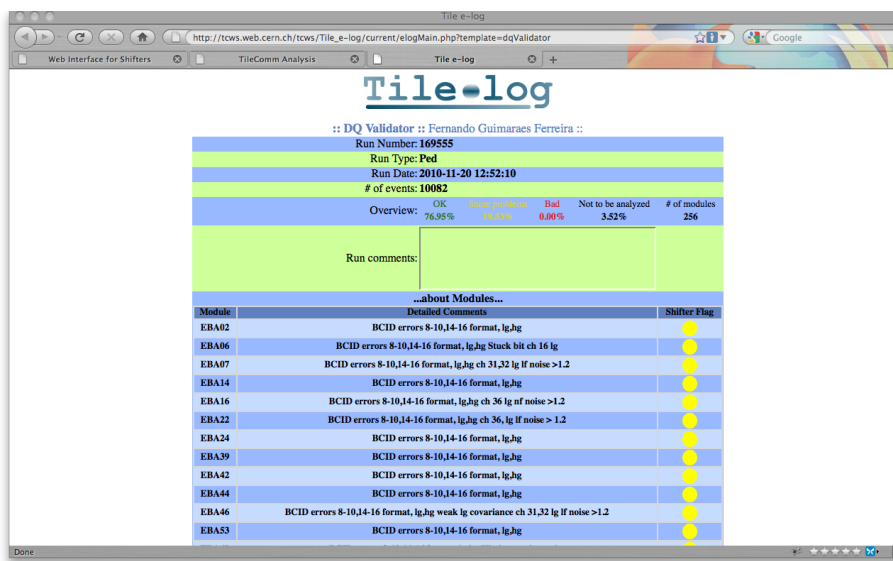


Figure 8. View where all the information about a run is summarized.

This view is responsible by concatenate all the information about the analyzed run for creating an entry in the collaboration's official electronic log (*e-log*). All the well behaved modules are ignored in this step, because only problems are interesting at this point. Also, the same overview displayed in the run list in the first view is submitted. The validator has to write down the finals comments about the whole fun. By submitting this information, the data quality validation is done for a run.

The second button in the top toolbar links to web system that displays data from the Tile Detector Control System (DCS) [4]. This system is responsible for providing enough information for the monitoring of the detector's power supplies behavior. For this specific case, the mean and standard deviation values for voltages and currents during the time period of the given run. The interface present a graphic representation of the barrels. Each module is divided in two, one for the mean and the other for the standard deviation analysis. Colors are given according to the result of comparing these statistics to the nominal values. Figure 9 shows the described system. This integration is important because many problems during the data acquisition may be correlated to power supplies' misbehavior.

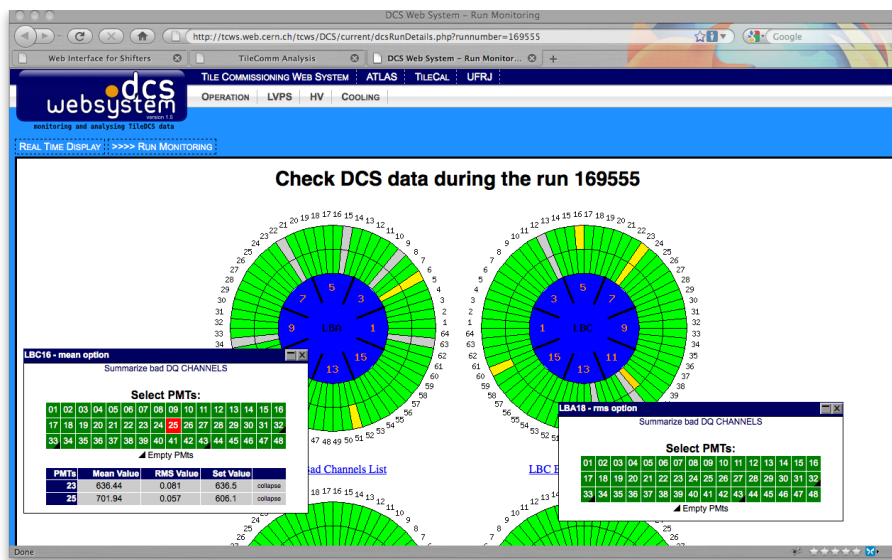


Figure 9. DCS Web System

The last two buttons are related to actions that the users commonly do. Frequently they give the same status for a whole run, for convenience, i.e., before starting an analysis, is common to attribute the green status for all modules since this will be, or at least it is expected to be, the most common diagnosis. In most cases, the user wants to hidden the modules that the DQM tool flagged as good. The DQM tool logic is pessimist, this means, any tiny problem, even the ones that do not prejudice the data acquisition, is noticed. In word words, if DQM says that a particular module is fine, it probably is.

5. Used Technologies

All views are based on dynamic HTML (DHTML), this is, combining HTML, Javascript and CSS elements [5].

Requests to the server and their response are made using messages in XML format. This communication was made in an asynchronous way, this means, not every request from the client results in a call for the server. Also, the client's interface doesn't hang waiting for the server's response. The AJAX method was used for handling this communication [6] and the JQuery implementation was the chosen one for developing the AJAX engine [7]. By doing this, the server load was hardly decreased and users got results in faster way. Multiple actions can be made at the same time and the server is protected from an overload of requests.

System's back-end (comprehends the controller and models patterns) is implemented mainly in Python program language. The system store its data into the official ORACLE database server.

6. Conclusion

During the ATLAS commissioning period several prototypes were purposed for supporting the offline analysis job. These prototypes became a set of web systems extensible used by the whole collaboration. The amount of data at that point allowed the use of a unofficial database without an accurate model.

It was necessary to redesign the system for dealing with the large amount of data produced during the ATLAS operation. The commissioning experience was very important for purpose the creation of an unique system for the whole data analysis process. A system capable of guiding the validator through the laborious job.

This way, a web system, using MVC software architecture, was developed. This system provides information on the latest runs, displays status from the automatic monitoring framework, presents the generated plots and stores the validation outcomes, assists to write logbook entries, creating and submitting the bad channels list to the conditions database and publishes the equipment performance history.

The System is accessible for the entire collaboration and runs under the Scientific Linux 5 environment in the AFS CERN's node.

7. Bibliography

- [1] ATLAS Collaboration. ATLAS: Technical Proposal for a General-Purpose pp Experiment at the Large Hadron Collider at CERN. Technical report, CERN, 1994. CERN/LHCC 94-43.
- [2] J M Abdallah, U Blumenschein, T Davidek, A Dotti, L Fiorini, C Maidantchik, C S Sanchez, and A Solodkov. Atlas tile calorimeter data quality assessment with commissioning data. *Journal of Physics: Conference Series*, 119(3):032001, 2008.
- [3] Anderson, K., Berglund, S., Blanchot, G., et al. Front-end electronics for the atlas tile calorimeter. Technical report, CERN, 1998.
- [4] C Maidantchik, F Ferreira, F Grael, and Atlas Tile Calorimeter Community. Tiledcs web system. *Journal of Physics: Conference Series*, 219(2):022029, 2010.
- [5] Jason Cranford Teague. *DHTML and CSS for the World Wide Web*. Peachpit Pres, 2001.
- [6] James Garret. Ajax: A new approach to web applications. article Adaptive Path <http://adaptivepath.com/ideas/essays/archives/000385.php>, 2005.
- [7] Bear Bibeault and Yehuda Katz. *Jquery in action*. Manning Publications Co., Greenwich, CT, USA, 2008.