

Integrated Electromyography Visualization with Multi Temporal Resolution

Pedro Cardoso*, Nuno Datia*, M.P.M. Pato*

*ISEL, Instituto Superior de Engenharia de Lisboa, Instituto Politécnico de Lisboa, Rua Conselheiro Emídio Navarro 1959-007, Portugal
Email: smeegoan@gmail.com

Abstract—For the analysis and comparison of electromyography (EMG) signals from different patients, standardization techniques are used to calculate the integrated EMG signal (iEMG), useful to evaluate the level of activity of different muscles. The iEMG corresponds to the area under the rectified curve (AUC).

Currently, monitoring and follow-up of these patients is done in regular health exams where the evolution of patients is assessed. The monitoring includes performing multiple clinical trials.

The goal of this paper is to help medical staff assessing the evolution of amyotrophic lateral sclerosis (ALS) disease by analyzing the collected data. The signal, described by the electromyogram, can be measured applying conductive elements or electrodes to the skin surface. The electrical activity of skeletal muscles is continuously measured for at least 24 hours.

In this work we used an appropriate data model to store data generated by EMGs, optimized for analytical processing. We implemented a Web API in order to provide access to the model data in an agnostic way, both to database management systems and data consumers. We implemented a web application to visualize data through the use of several interactive charts.

Usability testing helped to validate the solution, confirming the ease of use of the web application and the fulfillment of all the proposed goals. The telemonitoring ALS patients doesn't change mortality but reduces the need for hospitalization and costs for patient.

Keywords: Dimensional Modeling; Usability, OData; Data visualization.

I. INTRODUCTION

The *amyotrophic lateral sclerosis* (ALS) is a progressive fatal neurodegenerative disease with respiratory impairment. ALS is characterized by loss of spinal and cortical motor neurons that causes muscle weakness, paralysis and, ultimately, respiratory failure [4]. The first signs of ALS are: (i) muscle cramps; (ii) muscle weakness; (iii) cramps, and (iv) fasciculations.

There is no single diagnostic test for ALS [8]. Although, the disease is diagnosed by excluding other muscle and nerve disorders using a variety of clinical examination and series of diagnostic tests: *electromyography* (EMG), *magnetic resonance imaging* (MRI), genetic tests, muscle biopsy and others. After receiving a diagnosis, ALS patients attend regular

clinic visits about every 3–4 months¹. During these visits, each patient is monitored for changes in their clinical and functional status. Mean survival is about 3–5 years. However, approximately 10% of patients survive at least 10 years. This variable rate of disease progression makes prognosis difficult to predict and therapies challenging to develop.

This application is designed to complement the current monitoring method for patients who have been diagnosed ALS. Replacing the traditional tools such as spreadsheets with several interactive panels that allow filtering by various parameters of each patient, allowing the user to change the temporal resolution in which he wants to observe the exam(s), and provide the ability to compare the temporal evolution with the prediction of future results, as muscle failures, based on the history of previous measurements. This paper explores the technologies used in the creation of interactive graphics, to present data stored using a multidimensional model with a strong temporal nature. The study was conducted using, where possible, open tools and international standards.

This work aims to:

- 1) Implement an intuitive interface for end users;
- 2) Measure results and forecasts of future values through easy to read, accounting different level of detail;
- 3) Create a fluid application, fast data retrieval and instant response times. Even when the amount of information increases, the response time never exceeds the maximum time that users are willing to wait (about 2 seconds) [7];
- 4) Display the user a forecast of the evolution of muscular capacity of each patient, taking into account records of previous EMG;
- 5) Ensuring access control is applied to the patient data, because this is personal medical information.

To the extent of our knowledge, this is the first work that presents a historical vision of patients monitoring. In [9, 5] were developed a study design, follow-up of the patients and supervised data collection for ALS patients, but no with this proposal.

November 4, 2016

¹<http://www.als.net>

II. WEB API

The server component was implemented using a *representational state transfer* (REST) *web api* (WA).

A. Protocol

The REST WA supports *open data protocol* (OData) in order to meet at the same endpoint, clients with different needs. OData enables the server to return only the necessary information at any time, e.g. a subset of the properties of an object, providing mechanisms to filter collections, order sets of objects by specific properties, among others.

In the implemented solution, the entities provided by the WA, support only the methods of obtaining data (GET by key and GET for all entities). However, there is an entity that supports writing, used to store custom settings for each patient dashboard, through the usage of the POST, PUT and PATCH methods.

Serialization of the responses to each request is available in the following formats:

- . *extensible markup language* (XML) (Atom/AtomPub)
- . *javascript object notation* (JSON)

B. Security

Every time a client needs to access the *data model* (DM) it can only do it through the WA using an access token. This token is obtained using the specific application credentials and has to be passed on the HTTP header Authentication sent with every request. The obtained token has half an hour duration and, when it expires, the client needs to use the refresh token to obtain a new access token [12].

The clients authentication is made using an extension of the ASP.NET Identity, taking advantage of the already existing support to users, profiles and claims. This authentication infrastructure was extended, in order to support a bearer tokens authentication mechanism.

Because the *web client application* (CA) is hosted on a different location from the WA, and does several *asynchronous javascript and xml* (Ajax) calls in order to improve the data loading speed, the WA was configured to allow calls from absolute addresses different from its own.

The allowed addresses are the local server, for debug purposes, and the full url of the CA. The *world wide web consortium* (W3C) specification for *cross-origin resource sharing* (CORS) was used allowing *javascript* (JS) code to bypass the security policy imposed by browsers in order to access resources stored in a different domain.

C. Data Access

The DM is intended to store the *area under the curve* (AUC) patient data, obtained from the calculation of the EMG signal surrounding area. This model was implemented using SQL Server 2012 as the *database management system* (DBMS) and Entity Framework 6 as *object-relational mapping* (ORM).

III. CLIENT APPLICATION

A. Client Side

This layer is implemented in HTML, *cascading style sheets* (CSS) 3.0 and JS 1.8 and includes all the code that is run at the client side. In its implementation was sought to give priority to the initial page load. This was achieved using several different techniques like bundling, minification and asynchronous loading of required data.

1) *Measurements Prediction*: To help physicians in the diagnosis of patients, it is possible to enable the prediction of the evolution of AUC values. Thus, it is possible to show a hypothetical evolution of the patient's condition for a period of 12–24 months. The prediction is made by providing the measurements for the user selected time period. With the obtained equation, in the form $y = mx + c$, the calculation of the AUC is performed up to the end of the year following the last measurement on the patient.

2) *Charts*: The presented solution was implemented using the dc.js library. This library uses crossfilter allowing an exploration of multidimensional data. Internally, it uses the D3.js renderer to generate *scalable vector graphics* (SVG), data-oriented and responsive to user interaction in order to provide immediate answers. The EMG graphic rendering is done through the library dygraph. This choice was made to avoid transferring the necessary information to show the graphs all at once, when the facts are loaded. The dc.js library does not allow loading of partial data, and because only the last EMG for the selected time period is shown, the retrieval and rendering of this information is made only after the time filter interval is applied.

3) *User Interface*: In the development of the user interface several responsive web design techniques were applied at the layout level, allowing the site to adjust to any device or screen size [6]: The interface is designed to be accessible by as many people and devices through:

a) *Colors*: The graphics use pallets based on ColorBrewer² suitable for people suffering from color blindness;

b) *Languages*: The user interface is available in different languages. In the current implementation the CA Portuguese and English languages are available. The choice of language to be displayed is made respecting the HTTP header *Accept-Language* [1];

²<http://colorbrewer2.org/>

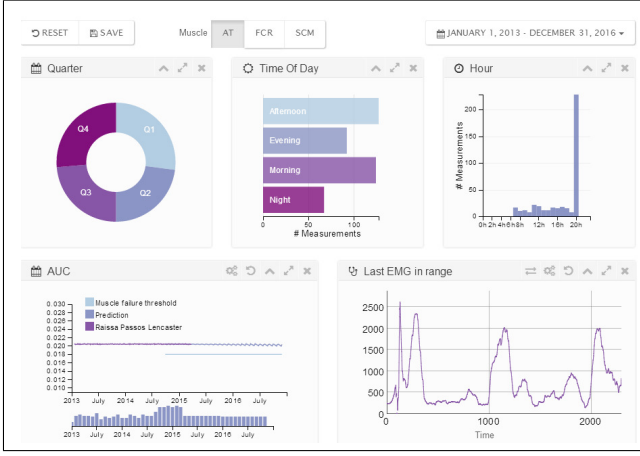


Fig. 1. Patient dashboard view.

c) *Images*: Font Awesome images are used because they are (SVG) and don't cause incompatibilities with screen readers;

d) *Customization*: The patient dashboard view, Fig. 1, is designed to be customizable and tailored to individual preferences, allowing the users to:

- . Apply and replace filters on the various charts presented;
- . Change the layout of the page by choosing the position of the graphics;
- . Enlarge the charts to see a single chart occupying the entire screen;
- . Choose one of several supported color palettes;
- . Persist the changes that have been made to the patient dashboard. The changes are saved individually for each user and appropriate patient summary;
- . Configure the way the AUC and EMG charts should be presented.

B. Server Side

1) *Security*: The data model of the CA is a subset of the WA authentication model, because it only uses the necessary tables to implement an ASP.NET Identity authentication provider, thereby allowing the use of the same database by both components, although this is not necessary.

2) *Communication with the Web Api*: The CA uses its own credentials to access the WA, regardless of which user is authenticated. This choice was made because it is not guaranteed that the CA and WA use the same database. Should the databases be different, the users authenticated in the CA would not be registered with the WA and as so wouldn't be able to authenticate and access the resources.

All communication made between the CA and the WA is made through data agents. Data agents are components responsible for communicating with the WA using the OData .NET library. The WA interface exposes the methods needed for the initial construction of the pages to show the user, leaving the fact retrieval for the client side.

Before the clients can access a protected resource of the WA they have to present a access token via HTTP Authorization header. Any entity in possession of a valid access token can use it to access the resources associated with that token.

The data agents are responsible for authenticating the CA with the WA in order to obtain the access token that will be used in each request. After obtaining the tokens, all subsequent calls of the client and server layers are made using the access token. When the access token expires it will be renewed making use of refresh token.

IV. APPLICATION PERFORMANCE

A. Test Environment

All measurements presented were taken using simple average in 3 different browsers. The system in which measurements were made has the following specifications: (i) Intel Core i7-4790 @3.6GHZ; (ii) 8GB; (iii) Windows 8.1 Pro; and, (iv) Microsoft SQL Server 2014 Express Edition (X64). The data provided with the implementation of this work was computer-generated and does not correspond to actual records. The average measurements per patient is 560. To make more noticeable the improvements made in loading times and also to perform a load test, 30000 patient records were generated with random values.

B. Data Normalization

The graphics library used, dc.js, supports data in the format *comma-separated values* (CSV) or JSON. The support to the JSON format requires that the data has a non-hierarchical structure. At this point the OData still does not allow to obtain data with this format, although it is planned [10]. Because the data obtained from the WA is in hierarchical format, it was necessary to implement a data conversion function to transform the original data into a compatible format with the graphics library used.

C. Data transfer

In order to reduce the amount of information transferred with each request we sought to optimize requests done to the WA. Taking advantage of the OData support of the \$select keyword it was sought to retrieve only the attributes on which the user will be allowed to apply dimensional filters. Thus the request to the original fact table was altered to indicate the attributes that are strictly necessary for the creation of graphics. This change prevented the transfer of redundant information, tests made found that the requests made by using the \$select keyword were about 38% of the size of the original request, Table I.

TABLE I
SIZE COMPARISON OF DATA REQUIRED TO CREATE THE DASHBOARD
BETWEEN THE ORIGINAL REQUEST AND THE REQUEST MADE BY USING
THE \$SELECT KEYWORD.

Request	Size (bytes)	Percentage
Initial	18122285	100
\$select keyword	6843985	37,766

D. Cache

After these optimizations we turned our attention to the model. Since we were joining everything together at the client side why not ask the server to send the data already formatted and ready to be loaded without having to do any preprocessing? To address this issue we implemented an indexed view that addresses the problem of initial data structure not being “flat”. This view makes the junction of the various dimensions of the fact table at the server returning results in a non-hierarchical structure.

Data from indexed views (also known as materialized views) are persisted to disk and require permanent storage as opposed to the simple views that are virtual and only take up virtual space at run time. This option significantly reduces the IO and CPU time, that the DBMS needs to perform the calculations thus providing performance gains [2]. Given the low volatility of the data present in the fact table and respective dimensions there are no negative consequences for the use of indexed views. The process that updates this data (its implementation is not in the scope of this project) can re-create the associated indexes whenever it needs to enter new data.

With the creation of the indexed view the data normalization became redundant. At this point the slowest component in obtaining data was the DBMS. To avoid the weight of making a call to the DBMS a caching mechanism was implemented. The cache was implemented in the WA and CA components allowing the change of its parameters, including its active state, through the `web.config` file of each component.

With this change we now have four possible model to access the data:

- . Model 1 — Measurements request that does not use the indexed view and transforms the data in a non-hierarchical structure at the client side;
- . Model 2 — Request measurements using the indexed view;
- . Model 3 — Request measurements using the indexed view and the WA cache;
- . Model 4 — Request measurements using the indexed view and the WA and CA cache.

At the outset one may think that applying cache in the CA would be unnecessary since we are already using cache in the WA, but this did not happen, Tables II and III. These results can be explained that with each request the CA has the weight associated with the network delay and the fact that each request must first be authorized by the WA before being

executed, which implies an access to the the DBMS in order to validate the access token.

TABLE II
TIME SPENT TIME IN MILLISECONDS FOR DIFFERENT BROWSERS TO GET
THE MEASUREMENTS OF THE FACT TABLE FOR A GIVEN PATIENT.

Browser	Model 1	Model 2	Model 3	Model 4
Google Chrome	6864	1898	1309	1202
Internet Explorer	7047	2028	1523	1250
Mozilla Firefox	6651	2012	1436	1241

TABLE III
MEASUREMENTS RETRIEVAL IMPROVEMENTS IN PERCENTAGE WHEN
USING DIFFERENT TYPES OF CACHE TO RETRIEVE A GIVEN PATIENT DATA.

Browser	Model 2	Model 3	Model 4
Google Chrome	72,35	80,93	82,49
Internet Explorer	71,22	78,39	82,26
Mozilla Firefox	69,75	78,41	81,34

V. TESTS

A. Usability Testing

In order to determine whether the proposed requirements had been met, a usability form was created. This form allows, through the use of iFrame, viewing the CA and the usability testing in the same browser window.

In order to avoid overloading users, and encourage the completion of the form, only 4 questions were chosen. The answer to the questions is made using a five-level 5 Likert scale. This scale allows users to indicate their level of agreement or disagreement on a symmetrical scale to a list of claims. It is thus possible to capture the intensity of the views of users on a given item [3].

The questions chosen for the test were as follows:

- 1) Access Patients section and click Help to view the functionality of the various components;
- 2) Open the dashboard of the patient Mauro Martins Fernandes;
- 3) Change the color scheme, change the location of dashboard panels “Time of day” with “Time”, minimize the panel “AUC”, maximize the panel “Last EMG”;
- 4) Find the last measured value of AUC for the sternocleidomastoid muscle in this patient;

Users who answered this form are mostly belonging to areas not related to health. In order to assess the ease of use and intuitiveness of the CA, none of these users had previous contact with the application or help in completing the form. We got 21 responses. Most of the answers were given by female users ($\frac{2}{3}$). The test was performed in different browsers, thus evaluating the proper functioning of the CA in different combinations of software hardware.

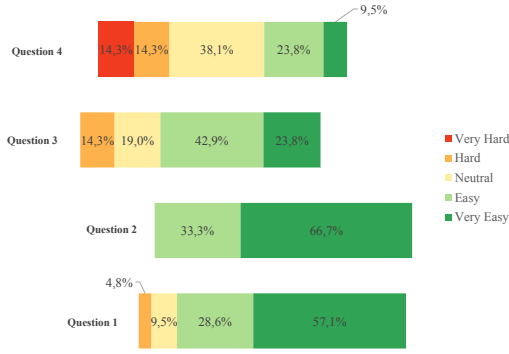


Fig. 2. Rating given by users to test questions.

B. Medical validation

To assess whether the implemented solution met the needs of health professionals, a demonstration of the CA was made in the Hospital Santa Maria. In the demonstration performed, the CA was praised by Professor Mamede de Carvalho and some viewing improvements were made:

- 1) Electric muscle signal — allow switching between the form or “envelope” of the EMG signal;
- 2) Oximetry measure — include other signals, e.g., measurement of peripheral oxygen saturation (SpO2).

The suggestion 1 was implemented through a low pass filter using the average value, in a signal sliding window [11].

The pseudo code for the envelope calculation code, through a rectify-and-mean approach in which the mean size of the window = T_w and time = N_w points is as follows:

$$y(t) = |x(t) - x_m| \quad z_{MovingAverage}(t) = \frac{\sum_{t-\frac{T_w}{2}}^{t+\frac{T_w}{2}} y(t)}{N_w}$$

This feature was added as a EMG chart alternative view. The user can choose the preferred view and set the size of the sliding window. The suggestion 2 was not implemented because the data available in the DM do not include the necessary information to show the oxygen level in the blood of each patient.

VI. CONCLUSION

Data visualization is key to understanding its contents. Being able to display large amounts of data to doctors or health professionals in a way they can understand and use is a major challenge.

The presented solution implements a DM best suited to the storage and retrieval of dimensional data. A WA was created that is agnostic to the DBMS using standard technologies avoiding proprietary protocols. This WA guarantees secure

data access to authorized users only. A CA was developed that, by making use of responsive technologies, is usable on different devices, allowing the analysis and visualization of patient EMG data in a interactive way.

Improvements in the performance of the developed components have enabled data load times of less than 2 seconds with a number greater than the expected records. Measurements were performed using approximately 30,000 measurements per patient. Through usability testing we proved the ease of use of the CA and its interface has been validated by a doctor, an expert in the problem domain.

REFERENCES

- [1] Roy Fielding et al. *Hypertext transfer protocol – HTTP/1.1*. Tech. rep. 1999.
- [2] Jes Borland. *SQL Server Indexed Views: The Basics*. <https://www.simple-talk.com/sql/learn-sql-server/sql-server-indexed-views-the-basics/>. [Online; accessed 14-June-2015]. 2014.
- [3] Alvin Burns and Ronald Burns. *Basic Marketing Research*. Second. Pearson Education, 2008.
- [4] Mamede de Carvalho and Michael Swash. “Can selection of rapidly progressing patients shorten clinical trials in amyotrophic lateral sclerosis?” In: *Archives of neurology* 63.4 (2006), p. 557.
- [5] Cinzia Ferrero et al. “Home clinical telemonitoring and domiciliar pneumologic medical assistance in amyotrophic lateral sclerosis ventilated patients”. In: *European Respiratory Journal* 46.suppl 59 (2015), OA4779.
- [6] Eva Harb et al. “Responsive Web Design”. In: (2011). <http://courses.iicm.tugraz.at/iaweb/surveys/ws2011/g3-survey-resp-web-design.pdf>.
- [7] Fiona Fui-Hoon Nah. “A study on tolerable waiting time: how long are Web users willing to wait?” In: *Behaviour & Information Technology* 23.3 (2004), pp. 153–163.
- [8] National Institute of Neurological Disorders and Stroke. *Neurological Diagnostic Tests and Procedures*. NIH Publication No. 05-5380. Online; accessed 2014 July. URL: http://www.ninds.nih.gov/disorders/misc/diagnostic_tests.htm. 2005.
- [9] Anabela Pinto et al. “Home telemonitoring of non-invasive ventilation decreases healthcare utilisation in a prospective controlled trial of patients with amyotrophic lateral sclerosis”. In: *Journal of Neurology, Neurosurgery & Psychiatry* (2010), jnnp-2010.
- [10] Mike Pizzo. *Consider adding support for requesting flattened results*. <https://issues.oasis-open.org/browse/ODATA-255>. [Online; accessed 2-May-2015]. Feb. 2013.
- [11] William Rose. *Electromyogram analysis*. Mathematics and Signal Processing for Biomechanics. University of Delaware. 2014.
- [12] IETF Trust. *The OAuth 2.0 Authorization Framework: Bearer Token Usage*. <http://self-issued.info/docs/draft-ietf-oauth-v2-bearer.html>. 2012.