



EyesTracker: Software para el procesamiento de registros electro-oculográficos*

Autor: Roberto Antonio Becerra García

Carrera: Ingeniería Informática.

Universidad de Holguín (Cuba).

Resumen: En este trabajo se aborda el desarrollo del EyesTracker, un sistema informático para el procesamiento de registros electro-oculográficos de movimientos oculares sacádicos, que calcula las variables de mayor significación clínica, conjuntamente con sus descriptores estadísticos, para el estudio de la ataxia, una enfermedad que tiene sus más altos índices internacionales de prevalencia en la provincia de Holguín. Este trabajo ha sido utilizado de forma satisfactoria durante más de un año, para el diagnóstico y evaluación del curso de la enfermedad y en ensayos clínicos a medicamentos candidatos para su tratamiento en el Centro de Investigación y Restauración de las Ataxias Hereditarias de Holguín.

Palabras clave: software; registros electro-oculográficos; movimientos oculares sacádicos.

EyesTracker: Software for processing electro-oculographic signals

Abstract: This work presents a computerized system for electro-oculographic recording of saccadic eye movements, known as EyesTracker. It calculates the variables that have a greater clinical significance together with statistical descriptors for ataxia investigations. Holguín has the highest rate of prevalence of ataxia in the world. The results of this research have been used successfully for over a year in diagnosing and investigating the disease and in clinical trials on candidate drugs for ataxia treatment at the Centre of Investigation and Rehabilitation of Hereditary Ataxia in Holguín.

Key words: software; electro-oculographic recording; saccadic eye movement.

Introducción

Cuba es el país que presenta la mayor concentración de enfermos con ataxias hereditarias a nivel internacional. La ataxia hereditaria espino cerebelosa tipo 2 (SCA2) es la forma molecular más frecuente encontrada en el territorio nacional, concentrándose la mayor cantidad de enfermos en la provincia Holguín, cuyas tasas de prevalencias e incidencias no han sido superadas por ninguna otra región o país a escala internacional (Velázquez, 2006).

Por esta razón, se crea en la ciudad de Holguín el Centro de Investigación y Rehabilitación de las Ataxias Hereditarias (CIRAH), encargado de realizar, entre otras contribuciones científicas, asistenciales y sociales, la instauración de un programa de neurorrehabilitación multifactorial, la caracterización neurofisiológica de los sistemas somáticos y autosómicos, la identificación de factores modificadores de la edad de inicio y el curso evolutivo de la enfermedad, la instauración de un programa de diagnóstico prenatal y presintomático para la SCA2 y el desarrollo de un modelo celular y animal transgénico para esta enfermedad.

Los movimientos oculares tienen un rol muy útil en la identificación de las disfunciones en un amplio rango de condiciones neurológicas, y entre estos, los movimientos sacádicos, utilizados para cambiar bruscamente el campo visual, proveen de una útil herramienta en la exploración de las funciones neurales.

A pesar de que los movimientos oculares se modifican por factores fisiológicos y patológicos, los fenómenos compresivos o traumáticos no suelen afectarlos como ocurre con las estructuras nerviosas periféricas. Por otra parte, las estructuras neurales que generan estos movimientos son blancos importantes de la ataxia tipo 2. Estas consideraciones convierten a los parámetros asociados a los movimientos oculares sacádicos en criterios de evaluación muy sensibles y de alto valor para el diagnóstico y seguimiento evolutivo de esta enfermedad.

Por esta razón resulta de gran importancia la detección de los puntos significativos de estas sácadas, con vistas a identificar su ocurrencia, y calcular los parámetros de significación clínica más importantes asociados a ellas.

También es necesario señalar que las afectaciones de esta enfermedad provocan que los registros oculares realizados a los pacientes presenten graves afectaciones en la forma de onda de los movimientos sacádicos, menores velocidades máximas, y la ocurrencia de movimientos involuntarios, parpadeos y temblores que modifican sustancialmente la forma de las sácadas, dificultando su identificación, la determinación de sus puntos significativos y el cálculo de los parámetros que las describen.

Para la obtención de los registros de los movimientos oculares de los pacientes en pruebas clínicas diseñadas específicamente con el fin de diagnosticar y evaluar la enfermedad se emplea en el CIRAH un equipo de fabricación alemana de la firma OtoScreen. Este equipo realiza las mediciones utilizando el método conocido como electro-oculografía, que consiste en la medición de los potenciales eléctricos en la zona cercana a los ojos, ofreciendo como salida un fichero ASCII, con los valores de los potenciales asociados a la señal de estímulo visual aplicada y la respuesta del paciente (García, 2007).

Este equipo no fue concebido específicamente para el procesamiento de registros de pacientes de ataxia, que están fuertemente contaminados por ruidos asociados. Tampoco permite la elección de criterios de determinación de los puntos de comienzo y

fin de las sácadás, ni se especifican los métodos de cálculo de un parámetro tan importante como la velocidad máxima sacádica.

La situación problemática gira en torno de que el equipamiento disponible para el registro de señales electro-oculográficas en el CIRAH no realiza la detección de los puntos de comienzo y fin de los movimientos sacádicos, de acuerdo a los protocolos aceptados en el CIRAH, ni realiza el procesamiento adecuado para la obtención de los parámetros de mayor significación diagnóstica y evolutiva en la ataxia SCA2.

Esto identifica el siguiente problema científico: ¿cómo elaborar una herramienta que realice el procesamiento de los registros clínicos de movimientos oculares obtenidos utilizando el equipamiento existente en el CIRAH, y calcule los parámetros y descriptores estadísticos adecuados para el diagnóstico y seguimiento evolutivo de la ataxia SCA2?

Con el fin de dar solución al problema antes planteado se define como objetivo general el desarrollo de una herramienta para el procesamiento de registros electro-oculográficos generados por movimientos oculares sacádicos.

Metodología

Para el desarrollo de esta herramienta se utilizó como metodología de desarrollo XP. Esta metodología, con un enfoque ágil, se adaptaba mejor a los requerimientos tanto del cliente como del equipo de desarrollo.

Se eligió el lenguaje de programación C++ por las características de rendimiento que este ofrece y como tecnología base las bibliotecas Qt. Estas últimas nos permiten poder liberar el producto final bajo la licencia de código abierto (LGPL) y compilar nuestra aplicación para las tres principales plataformas de escritorio en la actualidad: Microsoft Windows, Apple Mac Os X y GNU/Linux; permitiendo además que esta se adapte automáticamente a la cultura de cada uno de estos.

En la fase de exploración de XP se detectaron los siguientes requerimientos:

Funcionales

R1: Manejar estudios de registros electro-oculográficos.

R1.1: Crear estudio.

R1.2: Abrir estudio.

R1.3: Guardar estudio.

R2: Editar sácdas.

R2.1: Adicionar punto sacádico.

R2.2: Mover punto sacádico.

R2.3: Eliminar punto sacádico.

R2.4: Mostrar información de sácada.

R3: Generar reportes sacádicos.

R3.1: Generar reporte de prueba sacádica.

R3.2: Generar reporte de estudio sacádico.

R3.3: Generar reporte de grupo de estudios sacádicos.

R4: Detectar sácdas de forma automática.

R4.1: Detectar sácdas usando umbrales generales.

R4.2: Detectar sácdas usando umbrales por pruebas no funcionales.

Los requerimientos no funcionales son aquellos que constituyen propiedades o cualidades que el producto debe tener. Son las características que lo hacen atractivo, usable, rápido y confiable.

Apariencia o Interfaz externa

1. Uso de la cultura de cada sistema operativo en el que se despliegue la aplicación mediante el uso de bibliotecas que soporten esta característica.
2. Usar interfaces que se adapten a la resolución de la pantalla del usuario.
3. Uso de un tema de colores e iconos ligero y agradable.

Usabilidad

4. Realizar la internacionalización del software soportando los idiomas español e inglés.
5. Uso de atajos del teclado para facilitar la interacción con la aplicación.

Ayuda y documentación

6. Creación de manual de usuario del producto en un formato estandarizado.
7. Documentar las clases del API del núcleo de la aplicación utilizando un generador de documentación.

Portabilidad

8. La aplicación debe compilarse para las plataformas Win32, Linux x86 y Linux x86_64.

Personalización

9. La aplicación debe ser extensible y permitir su personalización mediante plugins.

Hardware

10. Una PC con un procesador de arquitectura x86 (≥ 2.0 Ghz de reloj).
11. Memoria RAM de 256 Mb o mayor.
12. Disco duro con al menos 50Mb libres.

Software

13. Sistema operativo de despliegue Microsoft Windows (win32) o GNU/Linux (i386 o AMD64).
14. Visor de PDF para la ayuda.

Diseño

Para el diseño de la solución propuesta se utilizó el estilo arquitectónico basado en componentes. Este estilo es el que mejor se adapta para implementar un sistema de extensiones basado en plugins, lo cual tiene como ventajas la extensibilidad de la funcionalidad y personalización e independencia del cliente a la hora de realizar el mantenimiento de esta.

Arquitectura del EyesTracker y sus componentes

La arquitectura propuesta para el EyesTracker presenta tres tipos principales de componentes: las bibliotecas, los plugins y las aplicaciones.

Las primeras permiten encapsular las funciones y abstracciones comunes de la aplicación divididos en subsistemas. EyesTracker cuenta con cuatro subsistemas:

Núcleo, Extensiones, Reportes y Visualización lo que se refleja en las cuatro bibliotecas elaboradas para implementar cada uno de estos que son respectivamente *etCore*, *etExtension*, *etReporting* y *etVisualization*. El prefijo *et* viene del acrónimo EyesTracker Library y sirve para especificar que estas bibliotecas pertenecen a este.

Los plugins son los componentes que permiten enriquecer la aplicación. La aplicación base del EyesTracker no está diseñada por sí sola para realizar ninguna funcionalidad en específico, sino como una plataforma que carga y despliega plugins para añadir nuevas funcionalidades. Hasta el momento se han implementado nueve plugins que responden a los requerimientos planteados por el cliente.

En el centro de todo se encuentra el EyesTracker, aplicación ejecutable que actúa como plataforma de plugins a través de las cuatro bibliotecas desarrolladas. Esto se puede observar con mayor detalle en el diagrama de componentes (Figura 1). Una ventaja que brinda este diseño es que la aplicación no trabaja directamente con los plugins, sino que utiliza las bibliotecas como puentes hacia ellos, lográndose una alta independencia de estos.

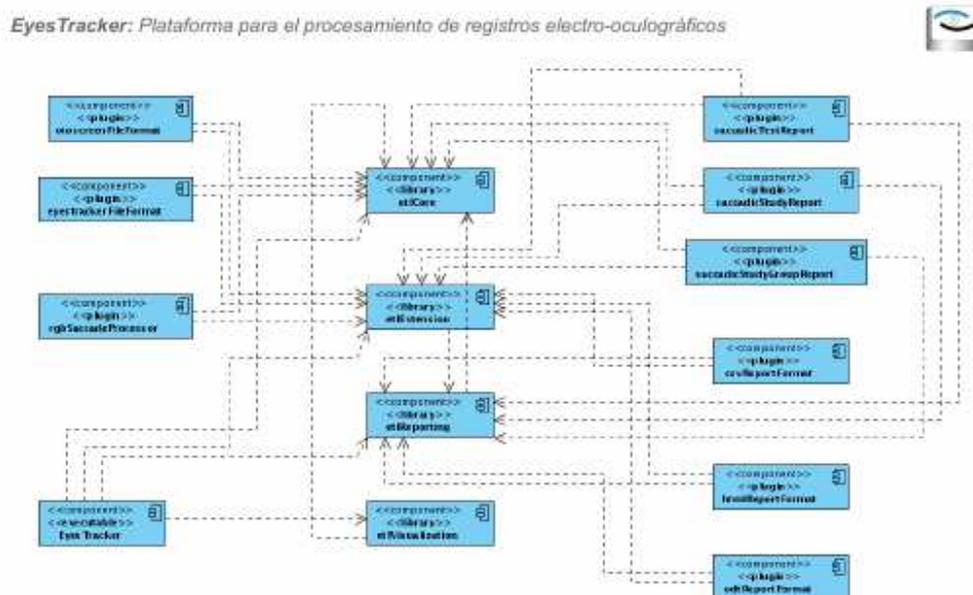


Figura 1. Diagrama de componentes y sus dependencias.

Es importante señalar que en todos estos componentes se hace un uso intensivo de las bibliotecas Qt, lo que hace que la base de esta arquitectura recaiga en las posibilidades de extensión de esta. Para organizar aún más las clases de estos componentes se utilizan paquetes (espacios de nombre en C++) para agruparlas. En la Figura 2 se

muestra la organización jerárquica de estos. Cada uno de estos pertenece a los siguientes componentes:

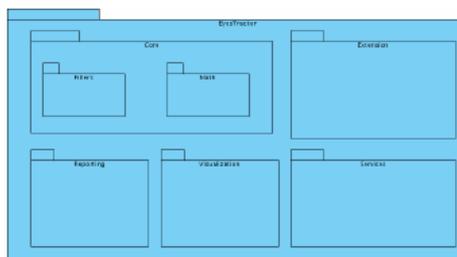


Figura 2. Diagrama de paquetes.

Bibliotecas

Las cuatro bibliotecas señaladas tienen la responsabilidad de crear las abstracciones y funciones comunes de la solución propuesta. A continuación se resumen sus principales funcionalidades:

- **Core:** Contiene las estructuras de datos de los estudios, pruebas, señales, artefactos de movimientos oculares y pacientes que se utilizarán en toda la aplicación. Además contiene filtros y utilidades matemáticas que facilitan la labor de procesamiento. Se considera el centro de la aplicación ya que no tiene dependencias y casi todos los componentes restantes dependen de su funcionalidad.
- **Extension:** Contiene los objetos que permiten cargar y manejar plugins. Además contiene la interfaz de plugin genérica y las cuatro interfaces de plugin específicas hasta el momento: Formato de Archivo, Procesador, Formato de Reporte y Reporte. Todos los plugins dependen de estas para implementar sus funcionalidades.
- **Reporting:** En esta se encuentran las estructuras comunes para todos los formatos de reporte ya sean títulos, tablas, textos, listas, etc. Estas estructuras son muy útiles a la hora de implementar nuevos formatos de reporte ya que para implementar un nuevo tipo de estos solo se necesita implementar a través de un plugin como se serializaría cada una de estas estructuras en el formato deseado. Esta puede ser usada en el futuro para construir un motor de reportes propio. También en esta biblioteca se encuentran otras estructuras que engloban reportes para las principales estructuras que se ofrecen en la biblioteca del núcleo como son la prueba y el estudio.

- **Visualization:** Esta biblioteca contiene el componente gráfico o widget para la visualización de señales y la edición de artefactos de movimientos oculares. Hasta el momento se encuentra implementado un editor de sácadas que permite tanto editar estas como mostrar sus datos en tiempo real.

Plugins

Como ya se había planteado, la solución propuesta cuenta con cuatro tipos de plugins. A continuación se caracteriza cada uno de estos:

Formatos de Archivo: Son aquellos que permiten importar de alguna fuente externa o almacenar estudios en un determinado formato. Actualmente existen para trabajar con el OtoScreen y para utilizar el formato propio del EyesTracker:

- **otoscreenFileFormat:** Importa registros generados por el OtoScreen.
- **eyetrackerFileFormat:** Crea, abre y guarda registros en el formato propio del EyesTracker. Permite además leer encabezados de este tipo de estudios para crear previsualizaciones de sus características.

Procesadores: Permiten el procesamiento de las pruebas que contienen los estudios, añadiéndole información a estos últimos. Hasta el momento se encuentra implementado uno para la detección de sácadas:

- **rgbSaccadeProcessor:** Detecta sácadas de forma automática en pruebas que las soporten. Lanza un diálogo donde puede elegirse los umbrales de detección a utilizar.

Formatos de Reportes: Permiten implementar nuevos formatos en los que se guardarán los reportes que se generen. En la actualidad existen para guardar en formato de texto tabulado Csv, Html y Odt:

- **csvReportFormat:** Exporta reportes a texto separado por comas. Muy útil para luego importar a paquetes de oficina como LibreOffice Calc y Microsoft Excel.
- **htmlReportFormat:** Exporta reportes al Lenguaje de Marcas de Hipertextos, además de dar soporte a la previsualización de este tipo de reportes por pruebas en la aplicación principal.

- **odtReportFormat:** Exporta reportes al Formato de Documentos Abierto (ODT). Este formato es el utilizado por la mayor parte de los editores de texto en paquetes de oficina libres.

Reportes: Generan reportes a partir de estructuras de la biblioteca del núcleo del EyesTracker. Existen para crear reportes de pruebas sacádicas, de estudios sacádicos y de grupo de estudios sacádicos:

- **saccadicTestReport:** Crea un reporte a partir de una prueba sacádica para su posterior visualización en la aplicación.

- **saccadicStudyReport:** Crea un reporte a partir de un estudio sacádico. Lanza un Explorador de Archivos para especificar la ubicación donde se desea guardar el reporte y su formato.

- **saccadicStudyGroupReport:** Crea un reporte a partir de un grupo de estudios sacádicos. Lanza un diálogo donde se especificará qué estudios se desean utilizar y qué variables se usarán. Muy útil para que los investigadores realicen análisis estadísticos de los mismos.

Aplicación principal

La aplicación principal es la encargada de cargar y manejar los plugins. Esta tarea es realizada mediante tres clases que sirven como fachada a los servicios que brindan los diferentes tipos de plugins con los que se cuenta. En la Figura 3 se observa un prototipo de la vista principal.

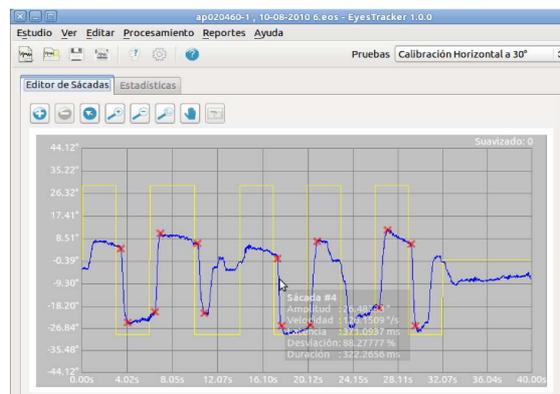


Figura 3. Vista de la aplicación principal.

Esta vista cuenta con varios tipos de diálogos que soportan la funcionalidad básica de la aplicación.

- **Ayudante de Creación de Estudios:** Sirve de guía en el proceso de creación de estudios.
- **Diálogo para Abrir Estudios:** Muestra una lista de estudios candidatos a abrir que se encuentran en una carpeta especificada. Además de permitir hacer búsquedas por nombre del paciente y por fecha de realización.
- **Diálogo de Configuración:** Permite configurar las rutas por omisión y el lenguaje de la interfaz de usuario.
- **Diálogo de Información del Paciente:** Muestra y modifica datos del paciente del estudio abierto actualmente.
- **Diálogo de Acerca de los Plugins:** Muestra los plugins cargados actualmente por la aplicación y sus propiedades.

Implementación y despliegue

Para la implementación de la solución se realizaron tres iteraciones del proceso de desarrollo de XP guiados por las siguientes Historias de Usuario (Casos de uso en RUP):

- Aplicación básica
- Gestión de estudios
- Editor de sácadas
- Generación de reportes
- Detección de sácadas de forma automática
- Creación de paquetes de instalación

Conclusiones

Se considera que los objetivos planteados al acometer la elaboración de este software fueron cumplidos. Esta herramienta de procesamiento de los registros clínicos de movimientos oculares ha sido concebida y desarrollada teniendo en cuenta el estado actual de la neurofisiología, en relación con la caracterización y evaluación de los movimientos oculares sacádicos, adaptada a las características del equipamiento, los usuarios y la información a ser procesada, y utilizando las tecnologías y herramientas de elaboración de aplicaciones informáticas de uso corriente internacionalmente, por lo

que responde a las expectativas y necesidades de los especialistas del CIRAH, y de otras instituciones médicas donde pudiese ser utilizada, en dependencia de la posibilidad de contar con el equipamiento necesario.

Recomendaciones

La solución desarrollada en el presente trabajo ha permitido llegar a una serie de conclusiones muy importantes para la continuidad del desarrollo de la misma y, a su vez, sirven de experiencia para proyectos que persiguen objetivos similares, hasta cierto punto. Por lo cuál se proponen las siguientes recomendaciones:

Crear nuevos plugins para dar soporte a otro equipamiento de registro de señales electro-oculográficas que no sea el OtoScreen.

Añadir las estructuras y los plugins necesarios para procesar otros movimientos oculares y no sólo a los sacádicos.

Soporte para añadir nuevos editores de artefactos mediante plugins.

Extender la aplicación de este trabajo a otras instituciones de salud y utilizarlo en otras patologías que afectan a los movimientos oculares sacádicos.

Referencias bibliográficas

VELÁZQUEZ, L. 2006: *Ataxia espino cerebelosa tipo 2. Principales aspectos neurofisiológicos en el diagnóstico, pronóstico y evaluación de la enfermedad.* Ediciones Holguín. Holguín.

GARCÍA, R. 2007: *Procesamiento de señales electro-oculográficas en estudios clínicos de Ataxia.* Tesis para la obtención del Diploma de Estudios Avanzados. Doctorado Cooperado en Informática. Universidad de Granada-Universidad de Holguín.

* Trabajo presentado en el XIX Forum Científico Nacional de Estudiantes Universitarios de Ciencias Técnicas. Tutorado por los doctores Rodolfo Valentín García Bermúdez y Fernando Rojas Ruiz.