

Software para Decodificación y Análisis de Archivos Binarios en Computadoras Navales

Mauricio J. Mancini^{1,2}, Christian L. Galasso^{1,2} y Miguel A. Banchieri¹

¹ Grupo SITIC - Dpto. de Ing. Electrónica, Facultad Regional Bahía Blanca, Universidad Tecnológica Nacional, Argentina.

² Servicio de Análisis Operativos, Armas y Guerra Electrónica, Base Naval Puerto Belgrano, Armada Argentina.

mancini_mauricio@hotmail.com, christian_galasso81@yahoo.com.ar,
mbanch@frbb.utn.edu.ar

Abstracto. En el presente trabajo se detalla el diseño de un software de decodificación de archivos y análisis de grabaciones de eventos navales. Originalmente, las embarcaciones contaban con un programa operativo de características similares, que se ejecutaba sobre las computadoras de 24 bits de abordo. La única interfaz que se disponía era una terminal tipo teletipo donde el operador insertaba comandos y se imprimían los resultados sobre un papel, el cual se utilizaba posteriormente para analizar los resultados del evento naval realizado. Debido a sus múltiples defectos y problemas, se decidió realizar una reingeniería del mismo para poder sustituirlo con un software que se ejecute sobre una PC de uso comercial, brindándole al usuario mayores herramientas y facilidades a través de las tecnologías disponibles en la actualidad.

Keywords: Decodificación, reingeniería, computadoras navales.

1 Descripción del Sistema Actual de Análisis

Las unidades navales, recepcionadas a partir del año 1982, poseen una serie de computadoras conectadas en red, las cuales conforman un sistema en tiempo real duro. La gran mayoría de ellas poseen la capacidad de realizar una grabación de las maniobras y datos actuales, denominado DataLogging (DL). Esta tarea genera un archivo binario, con un formato propio del sistema, en el cual se almacenan gran cantidad de datos correspondientes a sensores, movimientos de actuadores, estados y cálculos propios del sistema naval, durante el tiempo que se esté ejecutando esta tarea de DL. Los datos registrados varían en función de la práctica y la consola donde se realice la grabación, totalizando 13 posibles escenarios diferentes. Al finalizar la grabación, se crea un archivo, denominado COPYCASS, donde se encuentran todos los datos grabados de la práctica. Este archivo posee un tamaño variable que es

directamente proporcional a la duración de la grabación, pudiendo llegar a 6 horas como máximo (aproximadamente 16 MB).

1.1 Procedimiento Original de Extracción de un DL

Una vez que se tiene el archivo que contiene los datos grabados durante la práctica, se debe extraer para que sean analizados posteriormente por el personal correspondiente. Para hacer esto, se debe cargar un programa sobre la computadora de abordo denominado PFRD (Programa de Formateo y Reducción de Datos). Una vez ejecutado, se deben insertar una serie de comandos, dependiendo del tipo de evento y de los datos que se deseen extraer, por la terminal tipo teletipo de abordo. Terminado este proceso, se comenzarán a escribir diversos grupos de información, referente a conjuntos de datos específicos de cada sector de la embarcación. Se escribirá una fila completa de datos (entre 6 y 15 variables diferentes) por cada intervalo de tiempo grabado, una vez finalizado, escribirá otra fila en la parte inferior correspondiente al siguiente intervalo, siguiendo este proceso hasta terminar el total de la grabación.

La duración del proceso de extracción varía en función de la duración total de la grabación realizada, aunque este difiere según la práctica realizada ya que cambian los intervalos de grabación y la cantidad de datos mostrados sobre el papel. Para una extracción completa de uno de los eventos más comúnmente grabados, el tiempo de extracción de 15 minutos de grabación se puede extender a más de 8 horas. La duración total se torna tan extensa ya que se deben imprimir, para 15 minutos de grabación, aproximadamente 100000 valores de variables diferentes, todo a través de un teletipo que funciona de manera electromecánica y tiene obvias limitaciones respecto a la velocidad máxima de impresión que puede lograr.

El gran tiempo necesario para realizar una extracción no es la principal deficiencia que posee el sistema actual. Hay que tener en cuenta que durante todo el proceso, se debe dar alimentación al barco de 115 V 400 Hz a través de generadores impulsados con motores de combustión. Esto conlleva a que no solo el proceso de extracción de una grabación es muy extenso sino que además costoso.

1.2 Análisis de los Resultados

Una vez que se obtiene el resultado del PFRD impreso en papel, personal especializado se encarga de analizar estos valores para juzgar y calificar el desempeño de los operadores en sus respectivos cargos. Estos deben buscar valores específicos a lo largo de todos los datos impresos, para luego verificar los datos de otros parámetros en ese mismo instante de tiempo. Esta tarea es ardua y laboriosa ya que el hecho de buscar valores específicos en datos impresos sobre un papel es muy ineficiente y vuelve al proceso de análisis muy extenso.

2 Desarrollo de Aplicación PFRD Engine

Debido a las abundantes problemáticas que posee el actual sistema de extracción de DL, se planteó sustituirlo por una versión más moderna, la cual pueda ser ejecutada sobre una PC de tipo comercial y brinde herramientas que faciliten el análisis.

Para lograr el desarrollo de un sustituto, se debían realizar diversas pruebas al programa original, ya que no se posee el código fuente de este y la documentación disponible es escasa. Como primer paso se debía conocer el formato de los archivos generados por el sistema para identificar el modo en que se almacena la información. Luego se tenía que poder asociar los datos leídos a las variables mostradas por el teletipo descubriendo en qué posición se encuentra cada dato y de qué manera se convierten para ser mostrados al operador. Finalmente se debía desarrollar un software que se ejecute sobre la PC el cual debe analizar el archivo, extraer los datos, convertir las variables mostradas al operador y entregarlas de una manera más sencilla para su posterior análisis.

Se decidió denominar a este software “PFRD Engine” ya que es un motor de cálculo para convertir la información de un archivo de DL a valores de parámetros interpretables por el operador encargado del análisis.

El programa se desarrolló íntegramente en lenguaje C [1] [2], ya que brinda todas las herramientas necesarias y es fácil de implementar. El entorno de desarrollo elegido fue, el software libre, CodeBlocks [5].

2.1 Estructura del Archivo Generado

El archivo de generado (COPYCASS), se compone de distintos bloques de diversos tipos y tamaños, pudiendo ser: Volumen Header, Mark, File Header, Preset Block, Data Block, Last Block, File Trailer y Device Trailer. La estructura de los archivos sigue un patrón predeterminado (Fig.(1)), necesario para que el programa operativo pueda analizar sus bloques e interpretar sus contenidos.

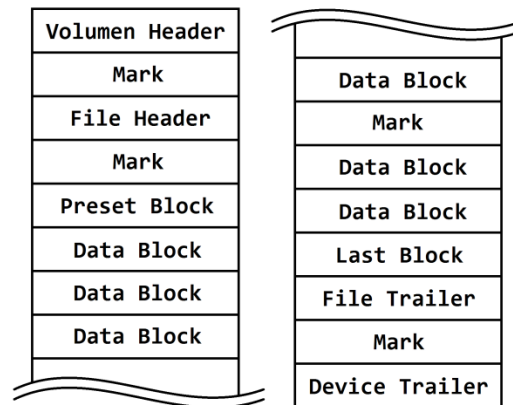


Fig. 1. En la imagen se puede ver la estructura del archivo COPYCASS segmentada en bloques de datos. Se identifica por el nombre a cada uno de ellos y se muestra el orden en el que estos se presentan normalmente en un archivo de DL. Se muestra el comienzo del archivo sobre la derecha y el final sobre la izquierda.

Como primer bloque de un archivo de DL, siempre se encuentra un Volumen Header. Este posee un tamaño de 255 bytes (85 palabras) de los cuales solo los 6 primeros contienen información. Comienza con los caracteres 'DL', seguido de cuatro números identificando al tipo de unidad naval. El resto del contenido no se toma en cuenta, aunque normalmente se completa con ceros. Seguido de este bloque se espera ver una marca (Mark). Dicho bloque solo posee dos bytes, los cuales están todos en 1 (0xFFFF). Se usa generalmente de separación, necesario cuando se leía en una cinta magnética por tiempos de aceleración y frenado. Como tercer bloque, se espera un File Header, el cual posee información más específica de la consola en la que se está realizando el DL. El tamaño de este bloque es de unos 81 bytes (27 palabras). Luego se vuelve a tener una marca sin información que sea de utilidad. En este punto suele aparecer un primer bloque denominado de Preset Block. Contiene información sobre el DL a realizar tal como, grupos, frecuencia de grabación de datos e información del sistema. Posee un tamaño variable, normalmente de 255 bytes (85 palabras), aunque en la primera palabra del bloque nos aclara su longitud. Llegado a este punto del archivo, es cuando aparece la información que se analizará posteriormente a través de unos bloques denominados Data Block. Estos contienen diversos grupos, dependiendo de cuál sea la práctica o ejercicio grabado. Dichos grupos son un conjunto de datos que almacenan información específica de cada subsistema en un intervalo de tiempo, tal como sensores, actuadores, rumbo, estados de la consola, otros. La longitud de los Data Block es usualmente de 255 bytes, aunque puede ser diferente en algunos casos, no obstante, en la primera palabra aclara cuál es su tamaño (Fig.(2)). Los diferentes grupos son almacenados en el Data Block hasta que complete su longitud predeterminada, pudiendo o no repetir un tipo de grupo dependiendo de la cantidad de

información que se grabe. De repetirse alguno en un mismo Data Block, significa que pertenece a un intervalo de tiempo diferente.

A partir de este punto se pueden recibir varios tipos de bloques en el archivo, usualmente son más Data Blocks cuya cantidad depende de la duración de la grabación. También se espera ver una marca cada 20 bloques de datos, solo a modo de separación. Por último, se lee un Last Block que nos muestra el final de la grabación del archivo. Se caracteriza por tener un tamaño de 255 bytes, siendo el primero un 1 y el resto de los 254 bytes 0's. Si se finalizó la grabación, el siguiente bloque es el File Trailer, de una longitud de 255 bytes de los cuales los primeros 3 contiene información mostrando los caracteres '///'. Luego de este bloque se espera otra marca a modo de separación. Finalmente el último bloque del archivo es un Device Trailer, con una longitud de 81 bytes, cuyos primeros 9 poseen caracteres indicando que se trata del final del archivo.

Data Block 1		Data Block 2	
Longitud de Bloque	Hora del Sistema	Longitud de Bloque	Hora del Sistema
Puntero al prox. Grupo		Puntero al prox. Grupo	
Grupo 5		Grupo 6	
Grupo 8		Grupo 3	
Grupo 1		Grupo 5	
Grupo 14		Grupo 8	
Grupo 6		Grupo 1	
		Grupo 14	

Fig. 2. En la imagen se puede ver un ejemplo de dos Data Blocks. En ellos se muestra cómo se organizan los datos en su interior separados por grupos. En el caso de la imagen, el *grupo 6* se termina de completar en el segundo bloque ya que completó el tamaño total del *Data Block 1*.

2.2 Decodificación del Archivo

El programa PFRD Engine, como primera tarea a ejecutar, realiza una lectura completa del archivo COPYCASS. Durante el recorrido del archivo se busca verificar que los bloques leídos correspondan al formato y orden de un DL. De encontrar algún bloque que no cumpla con el formato preestablecido, el programa detendrá su ejecución normal e informará al usuario el tipo de error y el número de bloque en el que se detectó. A medida que se leen los bloques, extrae su información útil y la

discrimina según corresponda. Como se mencionó previamente, los grupos poseen información específica de conjuntos de datos. Estos poseen un tamaño que varía según el grupo del que se trate y una estructura predeterminada. La primer palabra (24 bits) de cualquier grupo pertenece a un encabezado donde se dice el número de bloque y cuál es su longitud. A continuación, contiene datos de variables y parámetros, uno por cada palabra, aunque hay casos en que se utilizan variables booleanas y existen diversos estados dentro de una misma palabra.

Agrupar todos los datos de cada grupo por separado y se extraen datos de la grabación de los primeros bloques como el de Preset y encabezados. Finalizado este proceso, se posee cada grupo separado en orden temporal y listo para ser analizado.

2.3 Conversión de Datos

Teniendo los datos discriminados por grupos, se deben transformar a las variables mostradas por el PFRD. Para la extracción, el operador debe insertar unos nemónicos que simbolizan un conjunto de datos a extraer, como puede ser el conjunto de sensores del sistema (nemónico SEN), datos propios del barco (nemónico OSD) o datos de iniciales de la grabación (nemónico PRE), entre otros. En total existen 18 posibles conjuntos de datos para todas las grabaciones que se pueden realizar abordo. Los conjuntos obtienen la información que necesitan de grupos de datos específicos, uno o más, para luego transformarla y adaptarla.

La conversión de datos varía según el parámetro a mostrar, pudiendo ser una conversión lineal o a través de una operación algebraica. En las conversiones lineales, como se muestra en la Ec.(1), se divide al valor leído por una constante de conversión, que se relaciona con el tipo de unidad métrica del parámetro.

$$\text{resultado} = \text{valorLeido} / \text{ConstConversion} . \quad (1)$$

Las constantes de conversión se obtuvieron mediante diversas pruebas ya que no se encontró ningún manual donde se las mostrara. Inicialmente, se realizaban DL y se extraían por el método original para poder obtener las constantes de conversión. Hubo una gran cantidad de constantes que no se pudieron descifrar ya que muchos de los valores no variaban o se mantenían directamente en cero. Esto se debe a que los parámetros y variables de la embarcación rara vez tienen grandes variaciones para un periodo de tiempo corto, como puede ser el ejemplo de la temperatura ambiente o la presión atmosférica. Frente a esto, se decidió modificar manualmente las palabras de un archivo de DL y forzar a las variables a que tomen diferentes valores para observar la respuesta del programa original a cualquier tipo de entrada, incluso cuando esta tenga valores incoherentes.

$$\text{resultado} = \arctan(\text{valorLeido}_1, \text{valorLeido}_2) . \quad (2)$$

$$\text{resultado} = \arctan(\text{valorLeido_1}, \sqrt{\text{valorLeido_2}^2 + \text{valorLeido_3}^2}). \quad (3)$$

$$\text{resultado} = \sqrt{\text{valorLeido_1}^2 + \text{valorLeido_2}^2 + \text{valorLeido_3}^2}. \quad (4)$$

Otro tipo de conversión de datos se realiza a través de distintas operaciones algebraicas, como se muestra en las Ec.(2), (3) y (4). Estas son definidas en los manuales, aunque en la práctica se encontraron algunas diferencias y también el hecho de que las variables se deben escalar a sus respectivas unidades antes de realizar los cálculos pertinentes. En las ecuaciones se pone “valorLeido_X” haciendo referencia a una palabra específica de un grupo de datos predeterminado.

Se encontraron ciertos valores que no correspondían con ninguna conversión de las definidas previamente. En este caso se tuvieron que colocar mayor cantidad de valores en los campos de los grupos para tener mayor cantidad de muestras. Hacer esto de manera manual, editando el archivo binario, era inviable ya que pueden existir 16.777.216 valores posibles (2^{24}). Es por eso que se desarrolló un software que generara archivos de DL con los valores que se deseen. De esta manera se pudo descubrir el algoritmo conversión de ciertos parámetros. El software desarrollado también fue de gran utilidad para testear el funcionamiento del PFRD Engine, pudiéndole ingresar una gran cantidad de datos con combinaciones de valores que difícilmente de puedan replicar en una grabación convencional.

2.4 Interfaz de Usuario

Una vez que se convierten todos los valores a unidades físicas interpretables por el operador, se le entregan a través de un documento de Excel (.csv). Se genera un archivo por cada conjunto de datos (SEN, OSD, PRE, etc) que el usuario desee extraer. Dentro del documento, se muestran en las columnas los distintos parámetros y variables predefinidas para cada conjunto y en las filas los tiempos en los que se midió cada uno de estos valores (Fig.(3)).

Grupo SEN							
IDENT:	222222		INT:	660 MS			
TIME	VAR 1	VAR 2	VAR 3	VAR 4	VAR 5	VAR 6	VAR 7
SEC	SEC	Y	DEG	DEG	DEG	Y/S	Y/S
0.00	XXX.X	XXX.X	XXX.X	XXX.X	XXX.X	XXX.X	XXX.X
0.66	XXX.X	XXX.X	XXX.X	XXX.X	XXX.X	XXX.X	XXX.X
1.32	XXX.X	XXX.X	XXX.X	XXX.X	XXX.X	XXX.X	XXX.X
1.98	XXX.X	XXX.X	XXX.X	XXX.X	XXX.X	XXX.X	XXX.X
2.64	XXX.X	XXX.X	XXX.X	XXX.X	XXX.X	XXX.X	XXX.X
3.30	XXX.X	XXX.X	XXX.X	XXX.X	XXX.X	XXX.X	XXX.X

Fig. 3. Se muestra conceptualmente como se entrega la información al operador. En las columnas, se pone *VAR X* haciendo referencia a los parámetros y variables propias de cada grupo. Debajo de ellas se muestra el tipo de unidad, como por ejemplo *SEC* (segundos), *Y*

(*yardas*), *DEG* (*grados*), *Y/S* (*yardas por segundo*), entre otros. En la parte superior se muestra el *nombre del grupo* (*SEN*), el *identificador* de la grabación (*IDENT*) y el *intervalo* de tiempo de la grabación (*INT*).

Al entregarse los resultados en un archivo compatible con Excel (o similar), se busca brindarle más herramientas al usuario, aprovechando las propias del software que son ideales para análisis de gran cantidad de datos numéricos. Entre las herramientas que se pueden utilizar están la generación de gráficas, búsqueda de valores específicos, filtrado de datos y utilización de operadores matemáticos y estadísticos sobre un conjunto de datos. De esta manera, el usuario podría detectar el valor máximo, o mínimo, o promedio, de miles de valores en una fracción de segundo, siendo que por el método original debería hacerlo manualmente leyendo cada uno de estos desde un papel.

3 Testeo del Programa

Durante el desarrollo del programa, se lo sometió a una serie de pruebas [3] [4] donde se comparaban los resultados obtenidos con los que se mostraban en el PFRD original. Primero se crearon diversos archivos de DL donde se forzaban los valores. Se introdujeron gran cantidad de ellos para simular todas las posibilidades y observar como respondían ambos programas. Finalmente se confeccionaron planillas comparativas entre ambos PFRDs demostrando el correcto funcionamiento del nuevo software.

El segundo procedimiento de pruebas fue el de recolectar diversas pruebas de prácticas realizadas con anterioridad, para ser extraídas nuevamente pero esta vez con el PFRD Engine. Se volvieron a confeccionar las planillas mostrando los resultados de ambos programas, pero esta vez sobre archivos de prácticas reales abordado.

Superadas las dos pruebas de testeo se determinó el correcto funcionamiento del nuevo software y de su respuesta frente a cualquier tipo de archivo, ya sea con valores reales de una práctica, como de valores forzados que difícilmente se puedan encontrar en una grabación real o simulación.

4 Futuros Desarrollos

Actualmente se encuentra en desarrollo un nuevo software de análisis de archivos de DL. Este utiliza al PFRD Engine como motor de cálculo para obtener todos los valores ya convertidos, pero brinda al operador más herramientas que faciliten el análisis de las prácticas navales.

Se plantearon, en conjunto con el personal especializado, objetivos y funciones que se deben implementar en el nuevo desarrollo de software. Entre ellos se puede nombrar la búsqueda de valores específicos, navegación temporal del archivo,

capacidad de visualizar parámetros específicos de distintos conjuntos de datos o grupos facilitar las comparativas y la generación de archivos de tipo PDF protegidos para mostrar los resultados completos de manera segura y no a través de archivos fácilmente corruptibles.

Otro proyecto a futuro, es el de incorporar un software desarrollado previamente. Este programa estudia los eventos navales a través de videos tomados a bordo de la embarcación. Actualmente, posee una serie de errores o bugs que dificultan su utilización. Se planteó desarrollar un sustituto de este programa e incorporarlo al de análisis de archivos de DL que utiliza el PFRD Engine como motor de cálculo. De esta manera se podrían agrupar todas las herramientas de análisis de prácticas navales dentro de un mismo software.

Referencias

1. Darnell, P.A., Margolis, P.E.: *Software Engineering in C*. Springer, New York (1988)
2. Pagnoni, A.: *Project Engineering, Computer-Oriented Planning and Operational Decision Making*. Springer, Berlin Heidelberg (1990)
3. Stewart, G.: *Testing Safety-Related Software, A Practical Handbook*. Spring, London (1999)
4. Heitmeyer C.: *A Model-Based Approach to Testing Software for Critical Behavior and Properties*. In: Petrenko A., Simão A., Maldonado J.C. (eds) *Testing Software and Systems*. ICTSS 2010. Lecture Notes in Computer Science, vol 6435. Springer, Berlin, Heidelberg (2010)
5. Open source software CodeBlocks, <http://www.codeblocks.org>