TRABAJO FINAL
ESPECIALIDAD EN INGENIERÍA DE SISTEMAS EXPERTOS

UN MÉTODO DE PREPROCESAMIENTO DE DATOS ORIENTADO AL USO DE EXPLOTACIÓN DE INFORMACIÓN BASADO EN SISTEMAS INTELIGENTES

Autor: Lic. Hernán Merlino

Directora: M.Ing. Paola Britos

2004

ÍNDICE

CAPITULO 1 – INTRODUCCIÓN

Preparación de datos

La exploración y análisis, en forma automática o semi-automática, de grandes volúmenes de información para la detección de patrones de comportamiento es lo que se denomina minería o explotación de datos, también conocido por su vocablo en inglés data mining.

A grandes rasgos podemos definir el ciclo de vida de la explotación de datos como:

- Fase I: Obtención de datos a procesar.
- Fase II: Transformación de los datos para que pueda ser utilizado.
- Fase III: Aplicación de la técnica de explotación de datos.
- Fase IV: Evaluación de los resultados obtenidos.

este ciclo puede ser iterado desde la fase I, las veces que sea necesario adaptado al patrón obtenido para generar y detectar, de existir, un nuevo patrón de comportamiento; por lo que se genera la necesidad de nuevas transformaciones en los datos para poder satisfacer el nuevo modelo.

En este ciclo de vida, el paso que insume mayor tiempo es la *Fase II: Transformación de los datos para que puedan ser utilizados*, dicha fase lleva aproximadamente el 60% del esfuerzo de desarrollo. Sobre este punto se puede afirmar que la transformación de datos, es un trabajo sumamente tedioso y que pocas veces se puede automatizar totalmente, debido al desconocimiento de las combinaciones que se puedan llegar a producir en grandes volúmenes de datos, esto hace que el responsable de hacer la transformación de datos opte por utilizar gran cantidad de pequeñas aplicaciones inconexas para la transformación de los mismos.

Como conclusión se puede establecer que la aplicación de un método de preprocesamiento de datos orientado al uso de explotación de información basado en sistemas inteligentes, será de gran ayuda y agilizaría el ciclo de vida de la minería de datos.

Objetivo del trabajo

El presente trabajo tiene como objetivo el de describir las características principales que debiera tener un método de transformación de datos, detallando las características necesarias que deberá poseer un entorno de trabajo, o como se lo denomina en inglés framework, para la automatización del mismo.

Descripción del contenido del trabajo

En el capítulo 1: INTRODUCCIÓN se describe brevemente el ciclo de vida de la explotación de datos y se menciona el principal problema de la preparación de datos.

En el capítulo 2: DESCRIPCIÓN DEL DOMINIO se da una introducción al estado de la cuestión, detallando dentro del ciclo de vida de la explotación de datos los pasos que se encuentran relacionados con la transformación de datos.

En el capítulo 3: DISEÑO DE LA SOLUCIÓN se detalla una solución de software basada en la generación del marco de trabajo, se especifican y se detallan las características del mismo.

En el capítulo 4: MÉTODO DE TRANSFORMACIÓN se da la guía de trabajo que debería seguir para la automatización del proceso de transformación de datos, utilizando el marco de trabajo antes descrito, como eje de la metodología.

En el capítulo 5: CONCLUSIONES se plantean problema abiertos y futuras líneas de investigación.

CAPITULO 2 – DESCRIPCIÓN DEL DOMINIO

Introducción

En este capítulo se hace un breve repaso del ciclo de vida de la explotación , y se detallan los pasos a seguir en la transformación de los mismos.

Ciclo de vida de la explotación de datos

En ciertos casos el disparador de un proceso de explotación de datos es la detección de un problema y la necesidad de corregir ese comportamiento anómalo; en otros no es necesario observar nada anormal solo se aplica el proceso de minería para detectar patrones desconocidos. De ser este último el caso aplicado, los resultados obtenidos en la explotación de datos deben ser sometidos a un proceso de validación conocido como minería de reglas de negocio, o en su forma inglesa *business rule mining*, el cual nos permitirá validar o crear una nueva regla de negocio.

Las fases del ciclo de vida a seguir son:

a. ***Obtención de datos a procesar:*** Suele suceder que este punto siempre parece mucho más sencillo de lo que realmente es, algunos de los problemas que se suelen encontrar es la falta de acceso a los datos, ya sea por razones de seguridad o por no encontrarse disponibles, es decir los datos se encuentran resguardados. Si son cuestiones de seguridad de la información, una vez superada las cuestiones burocráticas, ya estaremos en condiciones de acceder a los mismos. En caso de que los datos se encuentran resguardado el primer problema al que nos enfrentamos, es obtener el espacio suficiente para recuperar los mismos, de estar en alguna base de datos también es necesario obtener los recursos para poder acceder a la misma. Con estos pasos cumplimentados, la próxima tarea a realizar es una primera revisión de los datos obtenidos para conocer sus características.

b. ***Transformación de los datos para que pueda ser utilizado:*** El primer paso para la preparación de datos es conocer el problema a resolver, o al menos hacia que objetivo queremos llegar, sin esto nos resulta imposible conocer los datos que debemos extraer. Por otra parte debemos conocer la forma en que se debe presentar la información al

modelo seleccionado para la explotación de datos, con estas dos precisiones se puede comenzar a recolectar la información y trabajar con ella.

Cuando se esta trabajando en explotación de datos, se están utilizando datos que representan hechos de la vida real, esos datos deben se preparados para que las herramientas de explotación puedan trabajar con ellas. La preparación de los mismos no es un proceso automático, por lo cual es necesario aplicar nuestro conocimiento para generar el conjunto de datos necesario para poder aplicar un modelo de explotación. Por lo antes dicho podemos definir como el principal objetivo de la preparación de datos es tomar información manipularla, transformarla y presentarla para que pueda ser procesada por un modelo de minería de datos.

Para conocer que transformaciones debemos realizar y como la debemos presentar nos debemos hacer dos preguntas fundamentales.

1. ¿Qué solución debemos obtener?
2. ¿Que técnica de explotación utilizaremos?

La primera cuestión la relacionaremos con la cantidad de información que deberemos manipular, y la segunda cuestión con el formato que le deberemos dar. Con los datos accesibles y hecha la primera revisión de los mismos los pasos comunes en la preparación de datos, se puede definir como:

1. *Enriquecer la información:* Luego de analizar la información y teniendo respuesta a las preguntas antes generadas, se plantea la posibilidad de agregar datos a los ya obtenidos, pues la información con la que se cuenta no cumple con todos los requisitos necesarios para poder generar un conjunto de datos que sea aceptado por el modelo.
2. *Obtener casos testigos:* Esto se puede convertir en un proceso muy tedioso, la obtención de estos casos testigo nos permitirán definir si el modelo al que lo vamos a aplicar es viable o no en relación al conjunto de datos que tenemos.
3. *Determinar la estructura de los datos:* Para poder entender este concepto es necesario definir el término *conjunto de datos,* este hace referencia a los datos que serán utilizados por el modelo de minería de datos para encontrar patrones. La estructura de datos hace referencia a la forma en que las variables se

relacionan unas con otras en los conjuntos de datos. Es en esta estructura donde se buscarán relaciones y patrones de comportamiento.

4. *Construir el modelo de entrada de datos:* Se puede decir que hasta este paso en lo que nos hemos centrado es en obtener y conocer la información disponible. En este paso lo que se determinarán los procesos que se seguirán para el modelado de los datos, entre los cuales podremos nombrar:
    - Normalización
    - Tratamiento de los valores nulo o vacíos
    - Detectar series (las mas comunes de tiempo)
    - Reducir el ancho de los datos, es decir la cantidad de columnas
    - Reducir la profundidad, la cantidad de registros.
5. *Inspeccionar los datos:* Una vez generada todas estas transformaciones, el minero de datos necesitan evaluar el resultado para poder determinar si de las transformaciones hechas al conjunto de datos lo hace viable para que el modelo elegido lo pueda procesar.

c. Aplicación de la técnica explotación de datos seleccionada: Luego de realizar todas las transformaciones se procede a modelar los datos en función de la técnica de minería de datos elegida. Dependiendo de la técnica, se ejecutarán una o varias ejecuciones con uno o varios conjuntos de datos.

d. Evaluar los resultados obtenidos: Con los resultados obtenidos de las ejecuciones del modelo, se centra la atención en detectar y poder comprender el resultado de los mismos. Esta tarea no es para nada sencilla e insume gran cantidad de tiempo, esto se debe en muchos casos a la complejidad de los resultados obtenidos. De este análisis es de donde se puede concluir, que los resultados no han sido los esperados, por varios motivos, la técnica no es la correcta para la solución del problema; otra posibilidad es que el conjunto de datos, no halla sido el adecuado, que se deba generar otro conjunto de datos, para validar los resultados obtenidos en la primera modelización; es por estas razones que el último paso sea comenzar con el ciclo nuevamente.

El modelado de la explotación de datos es un proceso de aproximación cíclica, el cual se debe ir mejorando a medida que se conoce mas de la información con la cual se seta trabajando. Es por esto que es necesario reiniciar el ciclo has que la información obtenida satisfaga el requerimiento que la produjo.

CAPITULO 3 – DISEÑO DE LA SOLUCIÓN

1. Introducción

En este capítulo se detalla una solución de software basada en la generación del marco de trabajo. Se especifican y se detallan las características de las distintas partes del entorno de trabajo.

2. Definición de una solución

En el proceso de transformación de datos basado en un entorno de trabajo podemos dividirlo en partes, por un lado la generación de un conjunto de programas que se encargaran de la automatización de la tarea de transformación esto es conocido, comúnmente con el nombre de *controlador de tareas*, y por otro un conjunto de programas los cuales harán pequeñas transformaciones sobre los datos, estos forman el conjunto de herramientas que debe proveer todo marco de trabajo.

2.1. Detalle para el diseño del controlador de tareas

2.1.1. Arquitectura abierta

a) *Origen del requerimiento:* Por la naturaleza del tema a encarar, preparación de información para la explotación de datos, no nos es posible conocer la secuencia de pasos a seguir para el proceso que va desde la captura de datos a la carga del modelo elegido.
b) *Solución sugerida:* Generar una interfase universal en la cual el creador de la tarea confeccione la secuencia de pasos a seguir. El controlador solo validará la finalización o no de cada una de esa tareas.
c) *Herramientas recomendadas:* Para la generación de una interfase estándar se recomienda la utilización de XML; esto se debe a su amplia aceptación, por la gran cantidad de herramientas que permiten generar documentos XML. Solo se proveerá un esquema para que el formato del documento sea valido y pueda ser interpretado por el controlador de tareas.

2.1.2. Múltiple Plataforma

a) *Origen del requerimiento:* Esto surge de la posibilidad de encontrarnos con proceso de minería de datos que corran en distintas plataformas
b) *Solución sugerida:* El lenguaje de programación seleccionado es Java, por satisfacer este requerimiento, y por lo difundido que esta.
c) *Herramientas recomendadas:* El estado del arte indica que Java es el lenguaje multiplataforma con mayor taza de crecimiento del mercado, agregado a esto el poseer todas las características de un lenguaje robusto.

2.1.3. Procesamiento en Paralelo y Distribuido

a) *Origen del requerimiento:* En función del tipo de problema que se debe abordar, y haciendo hincapié en la disponibilidad de acceso a los datos, es necesario poder ejecutar tareas en paralelo y sobro distintas computadoras.
b) *Solución sugerida:* Desarrollar el controlador de tareas para que maneje múltiples hilos de ejecución y sea capaz de ejecutar tareas sobre una red de computadores que utilicen el protocolo de comunicación TCP/IP. De esto se desprende la necesidad de implementar mecanismos de Mutex y Semaforos.
c) *Herramientas recomendadas:* Utilizar las características que el lenguaje de programación provee en forma convencional.

2.1.4. Manejo de error unificado 2

a) *Origen del requerimiento:* Una tarea esta constituida de determinada cantidad de pasos, estos pasos son en realidad otros programas, los cuales pueden estar escritos en cualquier lenguaje de programación.
b) *Solución sugerida:* Utilización de un documento XML para dejar constancia de los errores producidos durante la ejecución. Para esto se especificara un esquema para validar el mismo.
c) *Herramientas recomendadas:* Ya especificado.

2.1.5. Procesamiento en línea o en segundo plano

a) *Origen del requerimiento:* Al ser el proceso de minería de datos un de iteraciones continuas, se hace necesario poder dar prioridad la ejecución de tareas.
b) *Solución sugerida:* Utilizar la forma nativa en la que cada sistema operativo maneja la prioridad de sus servicios.
c) *Herramientas recomendadas:* Crear un programa que genere una entrada en los servicios del sistema operativo, esta deberá poder se modificada.

2.1.6. Mecanismo de agenda de tareas

a) *Origen del requerimiento:* En función del volumen de datos que se debe tratar, se hace necesario poder agendar tareas para las horas en que la carga de trabajo no es excesiva.
b) *Solución sugerida:* Utilizar el mecanismo de agendamiento de cada sistema operativo.

c) *Herramientas recomendadas:* Crear un programa que genere una entrada en la agenda del sistema operativo, esta deberá poder se modificada o dada de baja

2.1.7. Mecanismo de seguimiento de tareas ejecutadas

a) *Origen del requerimiento:* Poder seguir todos los pasos y totales de ejecución que ha realizado la tarea.
b) *Solución sugerida:* Generación de un documento XML, validara contra un esquema para su correcta aplicación.
c) *Herramientas recomendadas:* Ya antes mencionado.

2.1.8 Versionado de Tareas

a) *Origen del requerimiento:* Todas las tareas deben ser resguardadas, al igual que todos los esquemas y documentos XML, deben ser manejados de forma segura.
b) *Solución sugerida:* Utilización de un control de versiones en el cual se almacenaran todos los archivos sensibles.
c) *Herramientas recomendadas:* CVS, es el control de versiones recomendado en función de su gran difusión y confiabilidad.

2.1.9 Capacidad para poder crear distintos ambientes de trabajo

a) *Origen del requerimiento:* Como se ha mencionado anteriormente el proceso de minería de datos es un proceso de ciclos que se van afinando continuamente, de esto surge la necesidad de trabajar en distintos ambientes, desarrollo, prueba, producción.
b) *Solución sugerida:* Toda ruta de acceso a la aplicación debe ser relativa a una raíz, que será manejada en función de los perfiles de usuario.
c) *Herramientas recomendada:* Programa para el manejo de perfiles y seguridad.

2.1.10 Capacidad de ejecución con un juego reducido de datos

a) *Origen del requerimiento:* En función de la cantidad de datos con la que se deberá trabajar, debe permitirse hacer corridas de tareas en forma completa pero ejecutando solamente un set reducido de registros para cada paso.
b) *Solución sugerida:* Este será un parámetro en la tarea a construir.
c) Herramientas recomendadas: En el documento de configuración encargado de la tarea se podrá especificar si se desea trabajar con un set reducido de datos.

2.1.11 Capacidad de ejecutar una tarea desde un paso hasta otro

a) *Origen del requerimiento:* Como se ha mencionado anteriormente el proceso de minería de datos es un proceso de ciclos que se van refinando continuamente, sumado a errores en las corridas de las tareas, es necesario poseer un mecanismo para la optimización de los tiempos de ejecución de las tareas.
b) *Solución sugerida:* El controlador proveerá un mecanismo para ejecutar tareas desde cualquier paso y hasta otro paso especificado.
c) *Herramientas recomendadas:* En el documento de configuración encargado de la ejecución de una tarea se podrá especificar dicho mecanismo.

2.2 Características que no se contemplan en el diseño del controlador de tareas

a) *Rendimiento:* Como el administrador de tareas solo se encarga de ejecutar pasos, es decir programas, no se especifican hacer requerimientos de rendimiento mas allá del de una tarea de control.

b) *Seguridad:* El manejo de la seguridad de las tareas accesos a recursos, ya sea disco, conexiones remotas, acceso a la red, permiso de ejecución de tareas, recaerá en el creador de la tarea que deberá velar porque se le asignen todos los permisos necesarios para que la ejecución de la misma. Sobre los permisos de ejecución o no de una tarea el administrador de seguridad del ambiente será el encargado de asignar la configuración necesaria para que los usuario puedan ejecutar las tareas.

c) *Documentos necesarios para el sistema:* Se pasara a detallar los documentos que el controlador de tareas trabajara.

d) *Documento de la tarea a ejecutar:* En este documento XML se detallara la secuencia de pasos que deberá ejecutar la tarea. Así como puntos de bifurcación y ejecución en paralelo, puntos de unión de distintos caminos de ejecución. Para cada paso de la tarea se detallara:

- Nombre del paso a ejecutar.
- Nombre del programa a ejecutar en ese paso y ruta de acceso en el sistema de directorios.
- Nombre de archivos de entrada para ejecutar en ese paso y ruta de acceso en el sistema de directorios.
- Nombre de archivo de seguimiento para ese paso y ruta de acceso en el sistema de directorios.
- Nombre de archivo de error para ese paso y ruta de acceso en el sistema de directorios.
- Nombre de archivos de salida para ejecutar en ese paso y ruta de acceso en el sistema de directorios.
- Nombre de archivo de parámetros de configuración para ese paso y ruta de acceso en el sistema de directorios.

e) *Documento de ejecución de la tarea:* En este documento XML se detallara las características que se le desea dar a la tarea, algunas de ellas pueden ser: se hará una corrida completa o una parcial, desde hasta que paso ejecutar; ejecutar todo el set de datos o solo una parte de los mismos, cuantos entorno de ejecución, desarrollo, test, producción; que prioridad tiene la tarea, agendamiento de la tarea, etc.

f) *Documento de errores de la tarea:* En este documento XML se detallara las características del error producido durante la ejecución de la tarea, el mismo formato será para el controlador de tareas como para los programas que representan los pasos.

g) *Documento de seguimiento de la tarea:* En este documento XML se detallara las características para llevar todos los contadores de totales generados por el programa.

h) Documento de configuración de paso: En este documento XML se detallaran de ser necesario toda la configuración especial que debería tener un determinado paso para su ejecución.

2.3 Detalle de las herramientas que se utilizarán para las transformaciones

La descripción de este conjunto de programas no abarcara la totalidad de los programas que serán utilizados, solo se crearán los que por la experiencia se sabe que serán utilizados en la gran mayoría de las transformaciones.

a) Visor de archivos: La necesidad de crear un programa para poder visualizar el contenido de los archivos con la información a transformar es de fundamental importancia para el manejo de los datos. Se podría argumentar que un editor de texto cumpliría la misma función, pero un problema que tienen la gran mayoría de los editores de texto es que antes de mostrar la información paginan la totalidad del archivo para que la navegación sea mas rápida; en función de que podemos llegar a tener que procesar archivos de varios Gigas esta modalidad tardaría varios minutos.

b) Validador de formato: Esta herramienta nos permitirá, pasar un archivo con la estructura del archivo, validar la estructura del mismo. Otra tarea que realizará es la de asignar valores por defecto que también se especificarán el archivo de parámetros. Un ejemplo de esto será modificar campos con valor nulo por espacio.

c) Sort: Este programa se no se creará, se trabajar con algunos de los tantos programas especializado en ordenamiento de datos.

d) Modificador de formatos: Este programa tendrá la función de cambiar el formato de un archivo a otro en función de un archivo de configuración, este archivo se basar en XSLT, para procesar todas las transformaciones deseadas.

e) Reducidor de amplitud de registro: Este programa se encargara de reducir la cantidad de columnas que posee un registro, sin perdida de información, es decir que el nuevo registro obtenido representa al registro original. Esto se debe a que ciertos métodos de modelado de explotación de datos no están preparados para recibir gran cantidad de campos.

f) Reducidor de profundidad de registro: Al igual que lo expresado anteriormente, con la diferencia que este reducirá la cantidad de registros utilizados.

g) *Detector de series temporales:* Uno de los problema mas usuales en la selección de los conjuntos de datos es la posibilidad de tomar un grupo de registros que representen una serie temporal y esto perjudicaría el resultado del modelado.
h) *Creación de datos aleatorio:* Muchas veces se presentará la situación en que no se dispone de los datos para su prueba, es por esto que es necesario poder generar conjunto de datos para poder probar nuestro modelo.

A continuación se harán algunas consideraciones sobre los conjuntos de datos los cuales se ingresaran en el modelo seleccionado, cuando se seleccionan conjuntos de datos es recomendable la generación de dos conjuntos de datos conocidos como:

- ✓ Conjunto de datos de entrenamiento
- ✓ Conjunto de datos de prueba

El conjunto de datos de entrenamiento es esencial para construir el modelo, es con el cual el modelo tratará de encontrar patrones comunes entre los datos, este conjunto de datos es una representación del mundo real, como tal tiene un cierto grado de incertidumbre, también conocido en la minería de datos con los nombres de distorsión o ruido. Esta distorsión la podemos clasificar de dos maneras diferentes, por un lado, errores en los conjuntos de datos que hemos seleccionado, para lo cual se deberá generar correcciones y procesos acordes para solucionar los mismos; y por otro los errores inherentes al mundo real, los cuales no los podemos relacionar con los métodos para la captura de los mismos. Estos últimos no pueden ni deben ser corregidos. Este tipo de errores es a lo que comúnmente se lo conoce como ruido en el conjunto de datos.

Dependiendo de la metodología para el modelado, la detección del ruido puede ser muy importante, esto es pues nos dará una medida de la certidumbre del modelo obtenido. Aunque este proceso es puede convertir en muy tedioso, es fundamental poder conocer el grado de distorsión que tenemos en nuestro conjunto de datos.

El primer gran problema ante el cual nos encontramos en la detección del ruido en nuestro conjunto de datos, es la falta de patrones conocidos en la generación de ruido en conjuntos de datos, lo cual lo hace difícil detectarlo. De igual forma es válido decir que en ciertos modelos, el ruido es necesario y en ciertas ocasiones se lo agrega para que el modelo se adapte mejor al medio. Una de las formas para detectar el nivel de ruido, es la utilización de otro conjunto de datos, nos da una medida del ruido existente en nuestro conjunto de datos.

La otra finalidad que se persigue al utilizar otro conjunto de datos, conjunto de datos de prueba, es la posibilidad de medir la cantidad de individuos que conformarán nuestro conjunto de datos; pues al evaluar nuestros conjuntos de datos sobre el modelo el resultado no debería variar mucho, de ser así podemos deducir que nuestro conjunto de datos o es muy pequeño o es muy grande y en este último caso el componente de ruido esta generando la distorsión; es posible afirmar que el conjunto de datos de prueba, nos permite evitar entrenar el ruido.

2.4 Componentes para el controlador de tareas

2.4.1. Sistema experto para el control de tareas

Un componente importante al controlador de tareas, es la posibilidad de agregar un comportamiento inteligente a la ejecución de una tarea, de la experiencia del manejo de controladores de tareas y de procesamiento de grandes cantidades de información, se pueden determinar el siguiente patrón de comportamiento: *Imposibilidad de asegurar el formato de todos los datos que se deberán procesar,* esto se debe a que los datos no estas totalmente normalizados y sus fuentes son diversas, nos encontramos con datos incompletos, mal formateados, fuera de rango, etc. La forma habitual de solucionar esto es en cada programa, paso, de la tarea se escribe el código para manejar estas contingencias. Como es de suponer esto solo contempla algunas de las tantas posibilidades que se pueden dar y con el agregado de sumar complejidad a un programa que no debería tenerlo; es por esta razón que la construcción de un sistema experto para el control del la ejecución es recomendado.

Es el sistema experto que se propone construir tiene algunas características particulares, como lo son:

- La base de conocimiento contiene solo unas pocas reglas inferidas de la experiencia que se ha tenido de otros proyectos.
- La carga de esta base de conocimiento se realizará con la experiencia que se valla obteniendo sobre la implementación de este sistema de data mining, pues cada implementación es totalmente diferente.

Se darán algunos ejemplos de reglas que será común encontrar en el momento de puesta en producción del sistema experto:

- Manejo de valores por omisión o nulos.
- Mantenimiento predictivo

Dentro de las reglas que se cargaran en función de la experiencia con los datos y la tarea a realizar se podría citar:

- Detener el proceso de la tarea luego de determinada cantidad de errores.
- Ante determinado error que acción tomar..
- Tiempo máximo de ejecución de un paso dentro de la tarea

esta cuestiones solo se las puede conocer en el momento en que se enfrenta a los datos a procesar.

De lo antes mencionado se desprende una responsabilidad que será volcada sobre el sistema experto, ante la cancelación por expiración de tiempo o error cuales son los pasos a seguir, ante la imposibilidad del sistema experto en función de su base de conocimiento de poder solucionar el problema y no poder volver el comando al controlador de tareas para que se reinicie la tarea desde el último paso bien ejecutado, el mismo deberá invocar a un subsistema de escalamiento de alertas por fallas.

2.4.2. Subsistema de alarmas

Ante la cancelación de una tarea y la imposibilidad de su reinicio en forma automática, el control del proceso pasa a este subsistema, su función será: *Generar las alarmas necesarias para que un operador humano de solución a el inconveniente y se pueda reenganchar la tarea*

Esta alarma, envío de mail, aviso telefónico, etc. también tiene asociada un tiempo de respuesta máximo, en caso que este tiempo de respuesta es excedido y no se halla solucionado el problema, el sistema de alarma avisará a la segunda persona en la lista de alarmas para esa tarea, y vuelve a tomar el tiempo máximo de espera, así hasta que algún operador solucione el problema o se termine la lista de alarmas. De producirse este último caso el control será parado nuevamente el controlador de tareas para que de por finalizada la tarea.

CAPITULO 4 – MÉTODO DE TRANSFORMACIÓN

1. Introducción

A modo de guía de trabajo se darán los lineamientos a seguir en la transformación de datos en la minería de datos.

2. Detalle del método

El método propuesto ha sido llamado Método Unificado de Transformación (MUT), el cual es el resultado de la experiencia adquirida en el procesamiento de grandes volúmenes de información sobre distintas plataformas, desde equipos IBM 390 a redes de computadoras personales que poseen alguna de las distintas versiones de Microsoft Windows existente, pasando por el AS 400 y diversas versiones de Unix.

Es menester mencionar que en esta categorización de plataformas se hace necesario agregar a la nueva generación de sistemas de planeamiento de recursos empresariales, del vocablo en inglés *enterprise resourse plainning (ERP);* esto se debe a que se han vuelto tan complejos, que el usuario puede abstraerse del sistema operativo con le cual trabaja su computadora personal y solo trabajar dentro del entorno que le facilita el ERP, entre estos haremos referencias a modo de ejemplo a uno de gran aceptación en el mercado que es SAP.

2.1.Requerimientos para la aplicación de la metodología

El encargado de la transformación de datos debe tener conocimientos básicos sobre la notación que implemente del lenguaje unificado de modelado, de su vocablo en inglés *unified modeling language,* de aquí en mas UML, específicamente se hace referencia a los casos de usos y los diagramas de secuencia.

El explotador de datos conoce los formatos de los archivos con los cuales deberá trabajar, además del formato de salida obtenidos por el proceso de transformación de datos y tiene permiso a los mismos y son accesibles el conjunto de datos de entrada que deberá transformar para poder ingresar los datos al modelo de minería de datos; además se cuenta con espacio suficiente para el proceso de los datos, vale aclarar que esta metodología antepone la agilidad y velocidad de procesamiento en detrimento del espacio de almacenamiento de archivos intermedios. Esta

característica del método se basa en que el espacio físico de almacenamiento hoy en día es lo mas accesible y mas barato en comparación con recursos de memoria y procesador.

2.2. Descripción de la metodología

Conociendo las dos preguntas fundamentales para el proceso de transformación, es decir, donde estoy y ha donde quiero llegar, el método recomienda la aproximación gradual al objetivo final basado en un conjunto de pasos que se basan en:

a) Análisis de los requerimientos de transformación
b) Modelado de las transformaciones
c) Codificación
d) Pruebas
e) Evaluación
f) Nueva iteración.

El principal objetivo del método propuesto no es realizar todas las transformaciones en un solo paso, sino que se realizan pequeñas modificaciones a los datos, se realizara una prueba de regresión completa de lo hecho hasta ese momento y una vez evaluada la misma de ser satisfactoria se volverá a reiniciar el ciclo con la próxima transformación a realizar. A continuación se detallan los pasos mencionados anteriormente:

1. Fase de Análisis de los requerimientos de transformación:

- *Formato de archivo para el ingreso al modelo:* El primer paso que se deberá dar es el de recabar información acerca de que es lo que necesitamos obtener, es decir, conocer el formato de debe tener nuestro conjunto de datos para poder ser ingresado al modelo elegido para la minería de datos. En este paso se puede elegir cualquier método de los existentes en la ingeniería de software para obtener la información necesaria, el único requisito es, al finalizar este paso, es poseer la especificación detallada del formato de datos para el modelo. Cabe mencionar que es posible encontrarnos ante la posibilidad que la misma persona que se encuentra encargada de las transformaciones es la persona que ha definido el modelo de minería de datos, en tal caso solo se especificará el formato de archivo. Por practicidad en el presente trabajo se desarrollará la técnica de entrevistas.
  - Definición de la técnica de entrevistas: La entrevista es una técnica para elicitar la información detallada de un individuo. Se usa en la educción de sistemas grandes como

parte de técnicas de educcion de alto nivel. Es una técnica estructurada que requiere desarrollo de habilidades sociales, capacidad de escuchar y conocimiento de diversas tácticas de entrevista.

- Fases: Las entrevistas tienen cuatro fases: identificar candidatos, preparación de la entrevista, entrevista y actividades que siguen a la entrevista. A continuación, se ven en detalle:
  1. *Identificación de candidatos:* Normalmente, se comienza con la persona que ha autorizado o patrocinado el proyecto. El diagrama de la organización puede ayudar a identificar a otra gente; a los que conocen por que el sistema se está construyendo y a los que usarán el sistema. Además, hay que identificar a los que interactúan con los usuarios, pues una vez que este instalado el sistema las interacciones se pueden modificar.
  2. *Preparación:* Se hacen los arreglos necesarios con la gente que se va a entrevistar. Esto incluye:
     - Las reuniones hay que planificarlas por adelantado.
     - El material que se les va a proporcionar,
     - Objetivos de la reunión.
     - Recordar fecha de reunión
     - Preparar cintas de grabación.

     Además, hay que preparar, antes de la reunión, una lista de preguntas para el entrevistado:
     - Primero, se identifican las preguntas que ayuden a determinar requisitos. Se pueden plantear a partir de las ideas generales de la clase de sistema que se va a construir. Y, según se abren nuevas áreas de investigación, se necesitará más preguntas.
     - Segundo, se organizaran agrupando las preguntas por aspectos que se relacionen entre sí.
     - Por último, se decide cuánto tiempo se dedica a cada aspecto, y se prepara el material de grabación, si así se decide.
  3. *Protocolo de la entrevista:*
     a. Comienzo de la entrevista:
        - ✓ Lo primero, si no se conoce al entrevistado, se comienza con presentarse uno mismo.

- ✓ Luego, se revisan los objetivos de la entrevista, por que se esta aquí, que se hará con la información recogida, el tiempo asignado a los aspectos a tratar.
- ✓ Muchos requisitos son expresados en notación matemática o gráfica. Si es así, conviene introducir estas notaciones para asegurarse que se van a comprender

b. Guías generales: Hay que seguir una seria de estrategias para incrementar la calidad de la información recibida.
   - ✓ La primera respuesta a una pregunta no es necesariamente completa o correcta. Hay que explorar más respuestas para mejorar la comprensión. Algunas formas de hacerlo son: resumiendo, o ayudando a comprender y a educir generalizaciones o abstracciones de alto nivel; reconstruyendo las respuestas, traduciendo a palabras comprensibles para el entrevistador las respuestas, evitando así las malas comprensiones de la terminología especializada; y mostrando implicaciones, para que el usuario lo confirme al entrevistado.
   - ✓ Hay que ser escuchador activo de la entrevista, mirando al entrevistado tomando notas pero sin dejar de escuchar.
   - ✓ Permitir al entrevistado realizar preguntas, pero siempre manteniendo el control de la entrevista.
   - ✓ Siendo amables, manteniendo tranquilo al entrevistado.
   - ✓ Uso de técnicas de comunicación no verbal: estar atento al lenguaje del cuerpo del entrevistado que refleja su estado de ánimo.

c. Mantenimientos del proceso visible: De vez en cuando es importante hacer preguntas que ayuden a asegurar que el proceso va bien: sobre si el tiempo empleado en un aspecto es correcto, si se deja algo olvidado, etc. Por otro lado, también hay que asegurarse que el entrevistado comprende la racionalidad de las preguntas realizadas.

d. Tipos de preguntas:
   - ✓ De tipo general, “conoce el formato necesario?”. Se usan para establecer el contexto del sistema.
   - ✓ Preguntas abiertas. Animan a respuestas sin restricciones. Por tanto, es una forma de obtener mucha información.
   - ✓ Preguntas cerradas. Se fuerza al entrevistado a proporcionar detalles sobre un aspecto particular. Si se abusa de este tipo de preguntas,

respondiendo si/no, se corre el peligro de acabar dando la propia versión de los requisitos y no la del usuario. Es malo también, intentar anticipar la respuesta.

- ✓ Preguntas que elevan el nivel de la entrevista, cuando esta se pone demasiado detallada, como "cual es el objetivo de esto?"
- ✓ Preguntas de contexto, para cambiar de tema. Hay que estar seguro que el entrevistado comprende el nuevo contexto. Es malo cambiar frecuentemente de contexto. Pues incrementa la confusión y alarga la entrevista.

e. Comprobación de errores: Durante la entrevista hay que comprobar periódicamente los errores de comunicación. Algunos de los tipos de errores mas frecuentes son:
   - ✓ Errores de observación: ante el mismo fenómeno diversas personas pueden ver cosas diferentes.
   - ✓ Errores de recuerdo: la memoria humana es falible.
   - ✓ Errores de interpretación de palabras comunes para el entrevistador y entrevistado.
   - ✓ Errores de enfoque, si sendos participantes en la entrevista están discutiendo sobre un aspecto con diferentes niveles de abstracción.
   - ✓ Ambigüedades del lenguaje natural.
   - ✓ Conflictos que provocan que al final el entrevistador recoja su propia versión.
   - ✓ Hechos no verdaderos que proporciona el entrevistado, y que en realidad son opiniones.

f. Finalización de la entrevista: Las entrevista puede acabar por diversas causas; se han respondido a todas las preguntas, se ha consumido el tiempo disponible o aparece en el entrevistado. Una vez finalizada, hay que asegurarse de:
   - ✓ Disponer de cinco puntos para resumir y consolidar la información recibida, destacando los aspectos que requieren mas información.
   - ✓ Explicar las acciones a seguir
   - ✓ Agradecer al entrevistado el tiempo y esfuerzo dedicado.

g. Actividades que siguen a la entrevista: Después de conducir una entrevista hay pocas actividades a realizar. Fundamentalmente son:

- ✓ Producir un resumen escrito de la entrevista, reorganizando las ideas y consolidando la información relacionada. Ayuda a descubrir ambigüedades, conflictos o información desaparecida.
- ✓ Proporcionar al entrevistado una copia y pedirle confirmación que refleja lo intercambiado durante la entrevista.
- ✓ Si en la entrevista se generó información dependiente de la memoria del entrevistado, confirmarla con fuentes fiables.
- ✓ Finalmente, revisar los procedimientos usados para preparar y dirigir la próxima entrevista, y así mejorar el proceso en el futuro.

Como resultado de la aplicación de la técnica detalla para la educción de requerimientos basado en entrevistas, se obtendrá la especificación del formato de archivo, a modo de ejemplo se da una especificación de archivo de entrada al modelo.

Nombre del campo	Tipo	Valores permitidos	Valor por defecto	Obligatorio	Máscara
identificador de usuario	Numérico	[0-9999]		Si	9999
Nombre	Carácter(20)	[A-Z]		Si	
Apellido	Carácter(20)	[A-Z]		Si	
Domicilio	Carácter(50)	[A-Z]		Si	
Número	Numérico	[0-9999]		Si	9999
Código postal	Alfanumérico(8)	[0-9999] [A-Z]	“”	No	Z999ZZZ

Con el formato de archivo de ingreso al modelo de datos ya especificado, se abren dos caminos de acción:

1. *Origen de datos:* Comenzar a recabar la información necesaria para poder detectar el origen de datos para la creación del archivo solicitado. Los pasos a seguir para poder llevar a cabo esta tarea son:

- *Repetir la técnica de entrevista, para detectar el origen de datos*: En esta etapa de entrevistas, lo que se observa es que la cantidad de personas involucradas es mucho mayor de lo que uno a priori puede suponer. Entre las cuestiones a tener en cuenta podemos citar:
    - Se deberá entrevistar al administrador de la base de datos, para conocer la antigüedad de los datos que se encuentran en línea en la base de datos.
    - Otra cuestión a manejar con el administrador es determinar las posibles plataformas donde se encuentran los datos, de ser todas almacenadas en Bases de Datos, cuales y que versiones.
    - Otro punto es solicitarle el diagrama de entidad - relación (DER), para conocer la estructura de las tablas y sus campos.
    - De esto surgen dos implicancias, por un lado, en función de la cantidad que se encuentran en línea, se deberá entrevistar, al encargado del resguardo de los mismos, para conocer desde hace cuanto tiempo se tienen datos resguardados y su posibilidad de acceso. La segunda implicancia es, del análisis del DER, surgirán dudas sobre el origen de los datos esto hará se conserven entrevistas con los responsables de los diversos sistemas. Aquí será necesario realizar entrevistas grupales para resolver las inconsistencias propias de todo modelo de datos.
- *Acceso a la información:* En la medida que se detecte las fuentes de los datos, se deberán tomar todos los recaudos para poder acceder a los mismos, algunas de las cuestiones que se deberá resolver son:

    - ✓ Cuestiones referentes a la seguridad de datos, formalizar los pedidos de acceso a la información
    - ✓ Si los datos se encuentran resguardados, es necesario disponer del espacio para su recupero y calcular el tiempo que llevara esta tarea, que puede ser muy significativa.

  Se presentará a modo de ejemplo un formato de documento el cual contara con el origen de cada dato.

Nombre de Campo	Origen		Responsable	Resguardo	Accesible
identificador de usuario	Base de datos	DBCMP	Administrador de Base de datos	Si	Si
	Tabla	Maestro			
	Campo	Usuario			
	Tipo	Integer			
Nombre	Base de datos	DBCMP	Administrador de Base de datos	Si	Si
	Tabla	Maestro1			
	Campo	NombreUsu			
	Tipo	Carácter(40)			
Apellido	Base de datos	DBCMP	Administrador de Base de datos	Si	Si
	Tabla	Maestro1			
	Campo	ApeUsu			
	Tipo	Carácter(35)			
Domicilio	Base de datos	DBCMP	Administrador de Base de datos	Si	Si
	Tabla	Maestro2			
	Campo	Dom			
	Tipo	Carácter(100)			
Número	Base de datos	DBCMP	Administrador de Base de datos	Si	Si
	Tabla	Maestro2			
	Campo	NumUsu			
	Tipo	Carácter(10)			
Código postal	Planilla Calc.	Códigos		No	Si
	Equipo	Serv01			
	Raiz	D:/			

2. *Características del conjunto de datos:* Con el formato de archivo ya especificado, se volverá sobre el modelo de minería de datos seleccionado, pero con la finalidad de detectar los requisitos del conjunto de datos para su uso, es decir, cantidad de registros necesarios para su aplicación, cantidad de conjuntos de datos necesarios para su validación o entrenamiento. Los paseos a seguir para poder llevar a cabo esta tarea son:

    1. *Repetir la técnica de entrevista, para detectar el origen de datos*: En esta etapa de entrevistas es necesario recabar información sobre el modelo de minería a utilizar, esto servirá para poder generar los conjuntos de prueba, como ejemplo de esto se puede citar el caso de la utilización de redes de neuronas para el proceso de minería de datos, el mismo según el tipo de

red elegida necesitará, diversos conjuntos de prueba para su entrenamiento.

2. *Análisis del modelo requerido y el disponible:* Con la información recabada en los pasos anteriores Origen de datos y Características del conjunto de datos es necesario generar un hito en el proceso, para esto se debe validar el formato de archivo para el modelo de datos, de este paso pueden surgir las siguientes alternativas.

- ✓ Todos los datos se encuentran disponibles, alternativa mas optimista.

- ✓ Ciertos datos no se encuentran disponibles, por la razón que fuera, tanto la no existencia de los mismos, razones de seguridad, o simplemente por no poderse recuperar del resguardo de datos.

Ante esta situación se plantea la necesidad de tratar de conseguir los mismos de otro origen, a modo de ejemplo podríamos citar al Instituto de Estadística y Censo, Internet, proveedores, etc. Si la información esta disponible en algún otro origen vuelve a generar los pasos que se han definido en el origen de datos. De no ser así, se plantea una cuestión relacionada con el modelo elegido para la minería de datos. La cuestión a resolver en este punto es la decisión de cambiar el modelo en función de los datos que tenemos

Regla de escritorio 1:

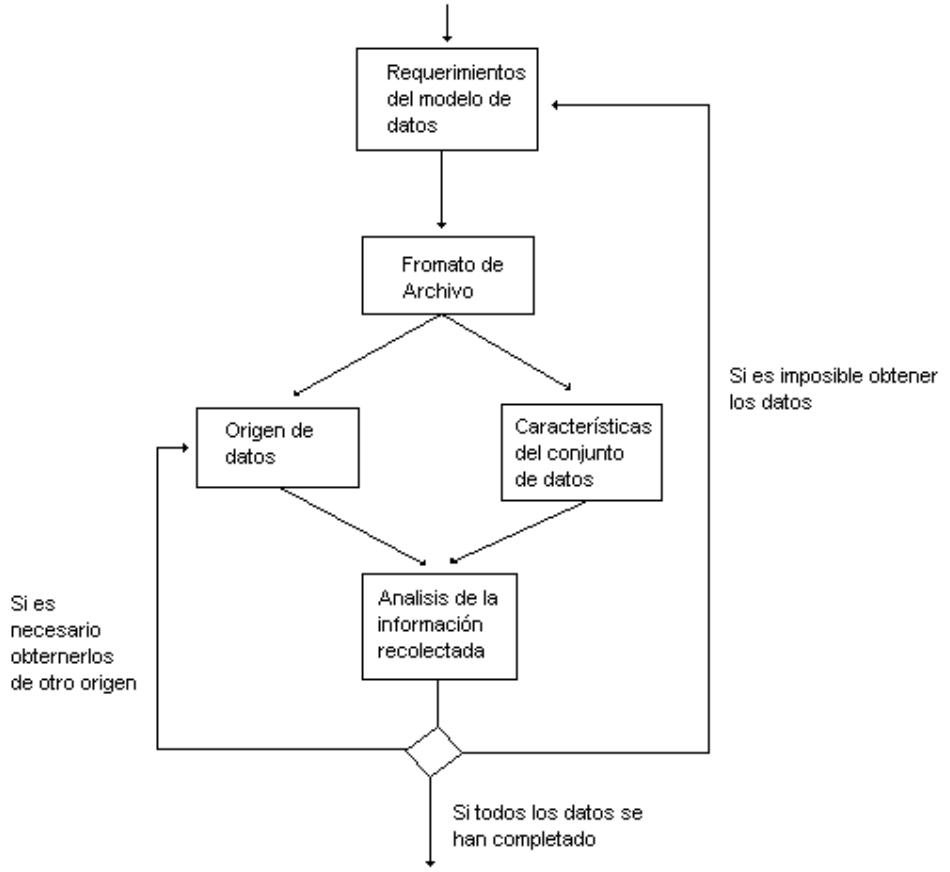
De lo mencionado anteriormente sobre la forma de obtener el formato de archivo y como se ha enunciado en el punto B, la detección de las características del conjunto de datos. Se podría pensar que ambas actividades se pueden realizar a un mismo tiempo, la experiencia en proyectos de minería de datos demuestra que, dentro del proceso de transformación de datos y dentro de ésta una de las etapas mas tediosa es la recolección de datos para formar los conjuntos de datos.

Regla de escritorio 2:

La decisión del modelo de minería de datos a utilizar debe estar en función del problema a resolver. En el caso que se nos plantee la situación, antes descripta, no poseemos los datos para proporcionarle al modelo, esto no debe ser considerado como un error a diferencia de esto el proceso de minería de datos, ha detectado el problema en una etapa temprana del desarrollo, que se puede definir como:

"El problema en cuestión esta dado por la falta de infamación en nuestros sistemas."

En el siguiente gráfico se esquematiza el proceso de obtención de requisitos para la transformación de datos.



2. Fase de Modelo de las transformaciones:

En esta etapa es donde se diseñaran las transformaciones necesarias para que los datos tomados del origen lleguen a la estructura requerida por el modelo de minería de datos.

Casos de Usos

Esto lo realizaremos utilizando Casos de Usos, se hará una breve reseña sobre que son los casos de uso. Los casos de uso, del vocablo inglés *use case,* constituyen el concepto principal del método OOSE de Ivan Jacobson, uno de los padres de UML. Los casos de uso representan, el medio para describir el carácter funcional de los objetos, son una representación orientada a la funcionalidad del sistema; permiten modelar las expectativas del usuario. Existen dos conceptos fundamentales en el modelado de los casos de uso:

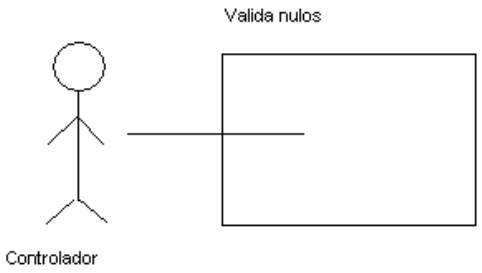
- ✓ Los actores que utilizan el sistema: Los actores pueden ser de dos tipos:
  - *b)* Humanos, usuarios de los programas
  - *c)* Software, programas que se comunican con nuestro sistema.

  Desde el punto de vista del sistema exista dos tipos de actores:
  - *a)* Los actores primarios: son los que utilizan el sistema
  - *b)* Los actores secundarios: tienen funciones de administración y mantenimiento del mismo.
- ✓ Los casos de uso que representan la utilización del sistema por parte de los actores.

La representación de los casos de uso puede ser textual o gráfica. Un ejemplo de una representación textual es:

Caso de Uso "Validar valores nulos"
• El controlador ejecuta el programa validar • El programa validar controla la no existencia de nulos • El controlador toma el control nuevamente • El controlador evalúa si se generaron errores

Un ejemplo de representación gráfica sería



Los casos de uso se pueden organizar desde mayor grado de abstracción hasta el detalle que se crea necesario.

Un escenario es una serie de eventos ordenados en el tiempo, que simulan una ejecución particular del sistema. De manera general, un escenario utiliza dos tipos de conceptos:

- ✓ Objetos que normalmente forman parte del sistema
- ✓ Eventos emitidos y recibidos por los objetos implicados en el escenario.

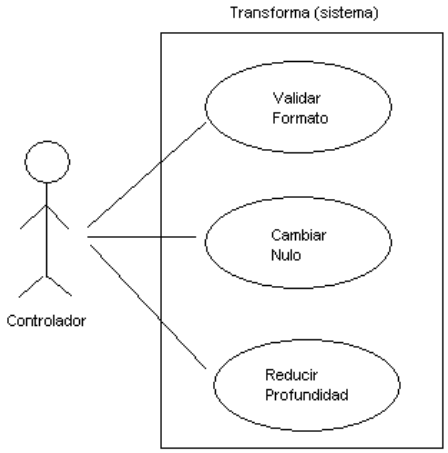
Los escenarios permiten experimentar las ejecuciones del sistema, por lo que resultan muy útiles para las pruebas y el mantenimientos.

Con las nociones básicas para modelizar las transformaciones se detalla las reglas a seguir para las mismas. En este punto ya se conocen las trasformaciones, pero se las debe considerar en forma separada, un concepto que debemos aplicar en esta metodología propuesta para su realización es:

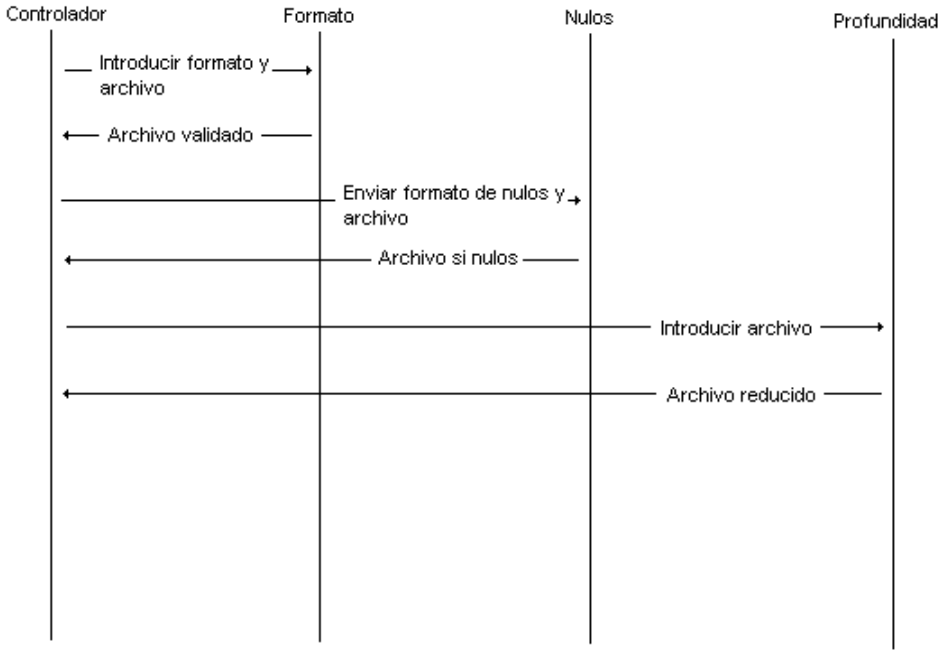
- o Es preferible utilizar varios programas especializados en las transformaciones que un solo programa que haga varias modificaciones. Esto se debe a que cuando trabajamos con grandes volúmenes de información el tiempo de procesamiento es un punto a tener muy en cuenta, de producirse un error en un programa único, el reproceso del conjunto de datos debería ser hecho totalmente, en cambio cada transformación hecha es procesada por un programa distinto de producirse un error, solo deberá reprocesarse el ultimo programa ejecutado.
- o Otra razón muy importante por la cual realizamos esta forma de proceso por partes es la probabilidad alta, de encontrar inconsistencias en los datos procesados; al ser procesados los datos por lotes, reducimos la posibilidad que estos errores se propaguen a todo el modelo. Aquí es donde se utiliza el controlador de tareas y las herramientas que integran el marco de trabajo.

El hilo conductor de todas las transformaciones que se deben realizar lo dará el controlador, en el es donde se configurará la secuencia de pasos, entiéndase por pasos programas encargados de realizar alguna de las transformaciones.

El modelado de las transformaciones tendrán como actor al controlador, que es el encargado de generar los eventos, para que el flujo de los datos tengan las transformaciones necesarias. A modo de ejemplo se dará un pequeño modelado utilizando casos de uso, escenarios y especificación de los requerimientos para el proceso de datos. El Caso de uso, donde se modeliza un proceso de transformación de datos, consta de tres operaciones básicas el primer paso es la validación del formato del archivo de origen, el segundo es el reemplazo de valores nulos por espacios en blanco y por último extraer de la totalidad de los datos disponibles un conjunto de datos, representativo del total. Se puede observar también que el actor de este caso de uso es controlador de tares este es el encargado de invocar a todas las tareas.



A continuación se modeliza el escenario sobre el caso de uso anterior.



En el ejemplo tratado, no es necesario profundizar los casos de usos ni el escenario de trabajo, para cerrar el mismo solo hace falta especificar el detalle de los formatos de entrada y salida de cada uno de los pasos involucrados.

Nombre de paso	Programa	Entrada	Salida	Observaciones
Validar Formato	formato	Archivo de origen	Archivo validado	Los archivos son de 1 giga. Ver performance.
		Archivo de formato	Archivo de errores	
		Archivo de errores	Archivo de pasos	
		Archivo de pasos		

Formato	Nombre campo	Tipo	Longitud	Obligatorio
Archivo origen	Id	Numérico	4	Si
	NomCli	Carácter	20	Si
	ApeCli	Carácter	20	Si
	Dir	Carácter	30	Si

En el modelado puede encontrarse la situación en la cual no se posea un programa en el marco de trabajo para satisfacer alguna de las transformaciones. Es aquí donde se pasa a la siguiente etapa de codificación.

3. Fase de Codificación

En este paso se codifican todos los programa que se necesiten para realizar las transformaciones necesarias para el modelo de minería de datos. El encargado de la codificación recibirá, al menos, las especificaciones de los formatos de entrada y salidas. Como el controlador de tareas es independiente del programa que debe ejecutar, se puede usar el lenguaje de programación que se desea, siempre y cuando este pueda ser soportado por la plataforma en la que se desea trabajar.

Cabe mencionar que se puede entregar al codificador toda la información que el encargado de las transformaciones crea necesario. Si el lenguaje así lo permite se podría entregar los diagramas de clases necesario para la codificación, así podríamos utilizar uno de los tantos esquemas que nos facilita UML.

Volviendo sobre la documentación mínima que se le debe entregar al codificador, en el cambo observaciones de la planilla con los nombres de los archivos que debe recibir y retornar el programa, es muy importante que el codificador sepa cuales son las principales características del archivo no ya de formato que las posee, sino de volumen de información pues ante distintas cantidades de información por procesar, la codificación será muy distinta. El codificador además de realizar el programa se encarga de hacer las pruebas de unidad de los programas que realiza, es pro esto que se le debe también facilitar un archivo de entrada con los datos reales, de ser posible con el volumen de información que en producción se enfrentará. Una vez que se ha finalizado con la codificación, el paso siguiente es la prueba de unidad y de regresión, por parte del encargado de la generación de la secuencia de tareas.

4. Fase de Pruebas

Esta etapa de prueba no solo se refiere a la comprobación de los programa encargados de la generación de las transformaciones, sino a la construcción del archivo que proveerá la secuencia de pasos al controlador de tareas. Con las primeras transformaciones a realizar, se carga el archivo de formatos del controlador de tareas, es necesario tener en cuenta que no es recomendable agregar varios pasos de una vez, lo recomendable es agregar un paso, hacer una prueba validar la salida y agregar otro paso. Las pruebas que se realizan son:

- *De unidad:* La finalidad de esta prueba es validar que el programa cumpla la función para la cual fue ingresado a la tareas, es necesario poder simular de la manera mas precisa posible una ejecución real. La validación que se hace es en función de la documentación antes desarrollada, se toman los formatos de archivo de entrada y salida, y simplemente se evalúa si los formatos son correctos.
- *De regresión:* En este tipo de pruebas lo que se debe realizar es la validación de los tiempos de procesamiento y recursos necesarios. Para realizar esto es necesario ejecutar la tarea completa hasta el ultimo paso que hemos agregado, es decir hacer una corrida completa de lo que tenemos hasta este momento. Lo que se busca probar es el tiempo de procesamiento, en la sucesión de pasos ejecutados es posible detectar que el tiempo de procesamiento es inaceptable para nuestro sistema, que los recursos utilizados son demasiados, etc.

De la evaluación antes descripta se pueden presentar distintas variantes:

- ✓ *Los tiempos son aceptables:* Esta es la posibilidad mas optimista de ser así, lo que se hace es continuar con el agregado de los siguientes pasos, esto puede ser que ya se tenga la especificación de la tarea y la próxima iteración a realizar solo sea agregar un paso mas y rehacer los ciclos de prueba.
- ✓ *Es procesamiento es demasiado extenso:* Esto hace que se deba replantear la estrategia de transformaciones a realizar, aquí se debe detectar cual es el paso que mas tiempo lleva y modificarlo.
- ✓ *Los recursos no son los óptimos:* Este tipo de alternativa se da cuando por ejemplo el espacio de almacenamiento intermedio es demasiado grande y no se dispone de mas espacio en disco, esto hace que sea necesario la reformulación de la estrategia a desarrollar.

Sobre los posibles caminos de acción que se puedan seguir en esta opción serán abordados en el próximo paso de la metodología propuesta

5. Fase de Evaluación:

Con toda la información de las pruebas antes realizadas el encargado de realizar las transformaciones, deberá tomar un camino de acción, como se ha dicho antes salvo que todo halla sucedido como se esperaba, en el resto de las opciones se deberá modificar algo.

La primera alternativa a seguir es una vez detectado el paso, programa, que mas recursos o tiempo demora, es tratar de optimizarlo. Otra alternativas no tan costosa es, la posibilidad de ejecutar las tareas en forma paralela, esto se hace agregando un punto de bifurcación en el controlador de tareas y se hace un procesamiento den paralelo; de no poder hacer esto un segundo camino de acción a seguir es la posibilidad es plantear generar nuevamente el programa en un lenguaje mas optimo, a modo de ejemplo podemos citar si se ha hecho el programa en un lenguaje como visual basic, se lo podría pasar a C/C++, para que su ejecución sea mas optima. Otra alternativa que podemos elegir es, a semejanza de la normalización de las bases de datos que en una primera instancia se normaliza, y para finalizar se realiza una desnormalización de las tablas para que estas posean una velocidad de acceso aceptable; se realizaran modificaciones en los programas que integran cada paso, para que se hagan mas de una transformación en un paso, como de la experiencia se ha observado que en cada paso los tiempos de acceso a disco, lectura del archivo y

escritura de los mismos, es lo que mas tiempo insume, unir transformaciones puede hacer que se reduzca el tiempo de procesamiento, aunque esto va en detrimento de los reprocesos que se puedan generar, en ciertos casos es la única alternativa mejorar la performance. Esto son algunos de los caminos alternativos que se podrán seguir para la mejora del rendimiento de la tara a ejecutar, en definitiva el encargado de la realización de las transformaciones tendrá la libertad de realizar las modificaciones que desee para poder llevar a buen puerta su trabajo.

6. Fase de Nueva iteración

De lo dicho hasta el momento se deduce la necesidad de generar nuevas iteraciones con cada paso de la tarea a realizar, este proceso se repite hasta finalizar todas las transformaciones necesarias para satisfacer el modelo de minería de datos.

De la metodología propuesta se desprenden algunas observaciones necesarias de hacer:

- ✓ Primero en función del método de trabajo el controlador de tareas es de suma importancia para la realización del trabajo, cuanto mas sofisticado sea el controlador y mas opciones pueda manejar, mejor será la forma que apliquemos la metodología.
- ✓ Sobre el uso del sistema experto, el controlador de tareas para que sea mas eficaz es necesario proveerle una herramienta, en este acaso un sistema experto, para que pueda manejar alternativas no contempladas por los programas hechos para generar los pasos de las tareas, podemos citar, cuando parar ante el primer error encontrado en el archivo o después de encontrar cien registros con error, y ante esta situación que se debe hacer enviar los registros erróneos a un archivo temporario para su posterior análisis. Ante estos errores que se debe hacer, en este caso una vez decidido que se ha producido un error el cual es la política que se debe llevar a cabo. En caso de ser necesario dar avisos, entra en juego el subsistema de alarmas, que es el encargado de disparar y controlar todas las alarmas que generara el sistema y el tiempo de respuesta de los mismos. Dentro de las tareas rutinarias en el proceso de transformación llevada a cabo una tarea que es necesaria es la evaluación de las ejecuciones. Esto se trata de generar un seguimiento de los procesos cuando ya se encuentran en producción, donde ya se ha automatizado la tarea y no es controlada por ningún operador humano.
- ✓ La experiencia dicta que todos los procesos con el tiempo se van degradando, su tiempo de respuesta empieza a ser peor, el espacio en disco utilizado aumenta, y esto puede llegar a niveles inaceptables, aunque el resultado final es el esperado.

Como se habrá observado en la metodología, cuando se detalla la documentación requerida para la generación de cada paso se especificó un "Archivo de pasos", el cual no se había tratado, este archivo es el cual nos permitirá evaluar el rendimiento de la tarea, en el mismo se contabilizaran los registros transformados , tiempo de procesamiento y demás información que se crea útil para el análisis del rendimiento en producción .

Lo que se realizara se podría denominar como una minería de datos del proceso de transformación de la minería de datos. Es ahí donde la utilización de un sistema experto puede tener mucho valor agregado, pues el mismo se encargará de analizar los tiempos de proceso compararlo con el volumen de información que se ha procesado y determinar un camino de acción a seguir.

Algunas situaciones que se pueden detectar con este análisis son, baja en la capacidad de procesamiento en determinados momentos del día, hay que recordar que nuestras pruebas aunque completas, no pueden simular todo el ambiente de producción donde otros procesos están corriendo en paralelo al nuestro y están compitiendo por los recursos del mismo.

Aumento de la cantidad de espacio necesario para la ejecución de los procesos, esto se puede producir por la fragmentación de la información en el disco, además de la perdida de respuesta, como se puede observar cuanto antes se detecten estos problemas menos traumática será su solución.

CAPITULO 5 – CONCLUSIONES

1. Problemas abiertos

La generación totalmente automática de las transformaciones de datos para su modelado en la minería de datos, es un aspecto muy importante por resolver.

Como hemos mencionado en el trabajo esta etapa es la que más esfuerzo requiere para su realización, poder llegar a un proceso automático redundaría en una considerable baja de costos y tiempo de implementación de una solución en la minería de datos.

2. Futuras líneas de trabajo

La utilización de un sistema experto que ayuda al trabajo del controlador de tareas, debería seguir desarrollándose para que el sistema experto sea un sistema embebido en el controlado, es decir que el grado de colaboración sea mucho mayor.

Sobre el subsistema de alarmas se debería migrar a un formato de sistema experto y embeberlo en el controlador de tareas.

3. Colorario

De lo expuesto hasta el momento podemos destacar que en el proceso de la minería de datos el modelo de minería elegido es solo una pequeña parte de la totalidad del proceso; este es uno de los errores más comunes que se encuentran en los proyectos de minería de datos, el de creer que lo único importante es obtener un buen modelo de minería de datos.

Esta idea la podemos extrapolar a gran cantidad de procesos donde esta involucrado algún mecanismos de inteligencia artificial, la no utilización masiva de AI en las empresas se debe a que solo se centran en el algoritmo y no en todo lo que rodea a ese algoritmo para que pueda ser utilizado, que a menudo insume muchos mas recursos y tiempo que el propio algoritmo.

REFERENCIAS:

Pieter Adriaans (1996). *Data mining.* Addison-Wesley

Dorian Pyle (1999). *Data preparation for data mining.* Morgan Kaufman

Michael J.A. Berry (1997) *Data mining techniques.* Wiley

Efrain Turban (1998) *Decision support systems and intelligent systens*. Printice Hall

Williman B. Frakes (1992) *Information Retieval*. PTR PH

Laureano F. Escudero(1977). *Reconocimiento de Patrones*. Paraninfo

Ivar Jacobson (1998). *The unifield software development process*. Addison-Wesley

Grady Booch (1998). *The unifield modeling language user guide*. Addison-Wesley

Grady Booch (1998). *Objects, Components, and Frameworks with UML*. Addison-Wesley

Grady Booch (1999). *The unifield modeling language reference manual*. Addison-Wesley

Craig Larman (2002). *UML yPatrones*. Printice Hall.