TecnoLógicas

Artículo de Investigación/Research Article

Una Aproximación Multi-Agente para el Soporte al Proceso de Extracción-Transformación-Carga en Bodegas de Datos

A Multi-Agent Apporach for the Extract-Transform-Load Process Support in Data Warehouses

Daniel Betancur-Calderón[1]
Julián Moreno-Cadavid[2]

Fecha de recepción: 02 de diciembre de 2011
Fecha de aceptación: 24 de abril de 2012

1 Escuela de Sistemas, Facultad de Minas, Universidad Nacional de Colombia, Medellín-Colombia, dbetanc0@unal.edu.co
2 Escuela de Sistemas, Facultad de Minas, Universidad Nacional de Colombia, Medellín-Colombia, jmoreno1@unal.edu.co

Resumen

Para brindar una solución adecuada en términos de robustez y automatización en el proceso de Extracción-Transformación-Carga (ETL por sus siglas en inglés) en bodegas de datos, en este artículo se presenta un modelo de sistema multi-agente que recopila las fortalezas de otros enfoques como son los wrappers y soluciones ad-hoc. Tal modelo considera la heterogeneidad y disponibilidad de las fuentes de datos, así como el carácter distribuido de los mismos. Para su validación se llevó a cabo una experimentación con datos tanto simulados como reales, la cual demostró no sólo su viabilidad técnica si no también su efectividad en cuanto a porcentaje de datos procesados y a tiempo para hacerlo.

Palabras clave

ETL; bodegas de datos; sistemas multi-agente; heterogeneidad; disponibilidad; automatización.

Abstract

In order to provide an adequate solution in terms of robustness and automation in the process of Extract-Transform-Load (ETL) in data warehouses, in this article a multi-agent model that gathers the strengths of other approaches like wrappers and ad-hoc solutions is presented. Such a model considers the heterogeneity and availability of the data sources as well as their distributed nature. For its validation an experiment was performed using simulated and real data, which demonstrated not only its technical feasibility but also its effectiveness in terms of the percentage of processed data and the time to accomplish it.

Keywords

ETL; data warehouses; multi-agent systems; automation; heterogeneity; availability.

1. INTRODUCCIÓN

ETL son las siglas en inglés de Extract, Transform, Load (Extraer, Transformar, Cargar), tres actividades dentro del contexto de bases de datos que al combinarse permiten el traslado de los datos de una ubicación (base de datos específica) a otra. Si bien este proceso puede tener múltiples finalidades, como por ejemplo migrar simplemente la base de datos, normalmente se lleva a cabo para alimentar bodegas de datos, las cuales son modelos de datos orientados a análisis donde los datos representan indicadores (medidas) que pueden ser observados de acuerdo a los ejes de análisis (dimensiones).

Como en cualquier proceso complejo, existen una serie de desafíos a la hora de realizar un proceso de ETL, siendo dos de los más significativos la heterogeneidad de las fuentes y la disponibilidad de las mismas. La heterogeneidad se refiere no sólo a que los datos de las diversas fuentes pueden estar en diferentes formatos lo cual obliga a realizar manipulaciones y combinaciones dentro de la actividad de transformación, sino también a que las fuentes como tal pueden ser de diferentes tipos. Algunas fuentes por ejemplo pueden referirse a modelos estructurados como bases de datos relacionales, mientras que otras pueden tratarse de modelos semiestructurados como hojas de cálculo, archivos CSV (siglas en inglés de Valores Separados por Coma), etc.; o incluso no estructuradas como archivos de texto sin formato, páginas Web sin contenido semántico, etc.

Uno de los enfoques más comunes para afrontar tal heterogeneidad es lo que se conoce como mediador-wrapper (Goasdoué *et al.*, 2000; Rousset & Reynaud, 2004), el cual se basa en mantener todos los datos en sus fuentes de origen y construir vistas de los mismos para satisfacer las consultas del usuario. Este enfoque funciona como un sistema centralizado donde el mediador realiza la integración de los datos de los que es responsable. Para esto reformula las consultas del usuario en función de los distintos contenidos de las fuentes de datos accesibles, utilizando un wrapper (traductor) por cada fuente. En otras palabras, el mediador se compone de varios wrappers, los cuales se encargan de llevar los

datos de las fuentes de origen a su destino, realizando una traducción cuando es necesaria.

Existen diversos trabajos que emplean este enfoque, por ejemplo el descrito por Zhou *et al.* (1996), siendo el presentado en Chawathe *et al.* (1994), conocido como el proyecto Tsimmis, uno de los más interesantes puesto que busca desarrollar herramientas que faciliten la rápida integración de las fuentes de información heterogéneas incluyendo datos estructurados y no estructurados. En éste trabajo se ubica un wrapper en cada una de las fuentes que convierte los objetos de datos subyacentes a un modelo común de información.

Una de las grandes ventajas de este enfoque es que permite acceder a los datos más recientes de las diferentes fuentes, sin embargo el hecho de que las consultas se realicen sobre demanda produce que exista cierta incertidumbre respecto a que las fuentes sean afectadas por algún cambio, es decir la eliminación total o parcial de las mismas, la modificación de los modelos de datos, etc. Otra característica de este enfoque es que posee una gran complejidad en la generación de consultas debido a que cada una debe ser reescrita para cada fuente, para luego realizar la recuperación, transformación y conciliación de los resultados arrojados.

Para resolver algunos de estos inconvenientes existen otros enfoques de tipo ad-hoc. En el trabajo propuesto en Simitsis *et al.* (2005) por ejemplo, se busca principalmente la optimización del flujo de trabajo de ETL por medio de algoritmos que se encuentran automatizados, logrando tiempos de respuesta satisfactorios. Otro trabajo relevante es el presentado en Vassiliadis *et al.* (2002), allí se muestra un modelo conceptual del proceso de ETL, que logra definir las actividades que se realizan, y proporciona fundamentos formales para su representación conceptual. Se resalta de este modelo que es personalizable y extensible de tal forma que el diseñador del proceso puede enriquecerlo. Otros trabajos realizados en encuentran en Viana *et al.* (2005) y Squire (1995).

Considerando las ventajas y limitaciones de los enfoques analizados y con el fin de atacar no solo el problema de heterogeneidad sino también el de disponibilidad de las fuentes de tal forma que se garantice que los datos cargados a la bodega de datos siempre sean correctos y estén actualizados, se propone en este artículo

una solución innovadora basada en el paradigma multi-agente, con la cual se pretende alcanzar un mayor nivel de automatización en las tareas involucradas.

Cabe aclarar que si bien ya existen algunas soluciones que emplean precisamente este paradigma (Boussaid *et al.*, 2003; Imtiaz & Hussain, 2005; Di Fatta & Fortino, 2007; Ding & Guo, 2009; Zhang & Ghen, 2010), se utilizó anteriormente el término "innovadora" porque tales trabajos o no abarcan la totalidad de las actividades que se consideran, o no describen en detalle la funcionalidad de las componentes (agentes de software) que las conforman, o no son validadas por medio de indicadores cuantitativos, como sí es el caso del trabajo presentado en este artículo.

La organización del resto del documento se encuentra de la siguiente manera: en la sección 2 se presenta una breve justificación del uso del paradigma multi-agente para el problema de estudio, así como una descripción del modelo propuesto; en la sección 3 se expone la validación de dicho modelo; y finalmente las conclusiones generales del trabajo se presentan en la sección 4.

2. APROXIMACIÓN PROPUESTA

Los Sistemas Multi-Agente (SMA) provenientes de la Inteligencia Artificial Distribuida (IAD) tratan sobre la coordinación inteligente entre una colección de agentes autónomos o semiautónomos, que existen dentro de cierto contexto o ambiente, se pueden comunicar entre sí y definen cómo pueden coordinar sus conocimientos, metas, propiedades y planes para la toma de decisiones o para resolver problemas complejos (Jennings *et al.*, 1998). En otras palabras, un SMA es un sistema distribuido en el cual los componentes son entidades dotadas con algún tipo de inteligencia artificial llamados agentes, o bien un sistema distribuido donde la conducta combinada de dichos agentes produce un resultado en conjunto inteligente.

Algunas de las características que en la literatura se suelen atribuir a los agentes y que son atractivas para la problemática analizada, descritas por autores tales como Franklin y Graesser (1996), son:

Autonomía: Un agente se basa en su experiencia para actuar en búsqueda de un objetivo individual o colectivo, sin requerir para ello del acompañamiento continuo de un usuario.

Sociabilidad: Este atributo permite a un agente comunicarse con otros agentes o incluso con otras entidades para entablar protocolos de cooperación o competencia.

Racionalidad: Un agente intentará siempre realizar lo "correcto" en términos de sus objetivos y restricciones a partir de los datos que percibe.

Reactividad: Un agente puede percibir los cambios de estado del entorno y actuar consecuentemente.

Partiendo de estas características, la solución que se propone en este artículo es un modelo basado en una arquitectura Multi-Agente para realizar el proceso de ETL. El objetivo de este esfuerzo es desarrollar un modelo flexible que además aproveche algunas de las bondades del enfoque orientado a mediadores y otras soluciones ad-hoc. El modelo presentado no está especificado para un dominio en particular, y busca trabajar tanto con datos estructurados como semi-estructurados.

Para dicho modelo se definen un grupo de agentes que buscan cubrir las principales tareas que se realizan en el desarrollo de las tres actividades de interés. Dichos agentes son responsables de acceder a las diferentes fuentes, recolectar datos y aplicarles algunas transformaciones para luego llevarlos a un esquema unificado dentro de una bodega de datos. En la Fig. 1 se muestra la arquitectura del SMA cuyos componentes principales son; los agentes recolectores, un agente integrador, un agente coordinador y una agente interfaz.

Los agentes recolectores son los responsables del acceso y extracción de los datos en una fuente de origen específica. Este agente debe ser dotado de un conocimiento de dicha fuente que le permitirá tanto acceder a ella, como también capturar únicamente los datos que sean relevantes. Tal conocimiento es encapsulado dentro de un XML, tal como se presenta a manera de ejemplo a continuación.

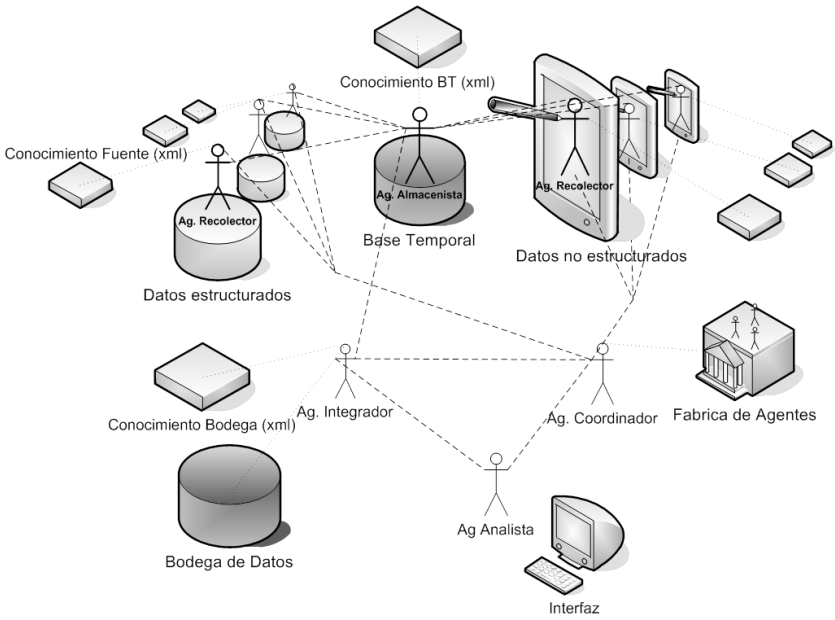

Fig. 1. Diagrama de despliegue del sistema. Fuente: Autores

```
<?xml version=" 1.0 " encoding=" UTF-8 " standalone= "yes" ?>
<fuente>
        <ID>1</ID>
        <nombre>Ventas</nombre>
        <tipo>
                <general>estructurada</general>
                <aplicacion>Oracle</aplicación>
        </tipo>
        <acceso>
                <ubicacion>
                        <general></general>
                        <direccionip>10.1.12.241</direccionip>
                        <puerto>8086</puerto>
                </ubicacion>
                <autenticacion>
                        <tipo>1</tipo>
                        <usuario>ventasCID</usuario>
```

```
                    <contrasena>ventas2009</contrasena>
                </autenticacion>
        </acceso>
        <concepto>
                <nombre>producto</nombre>
                <atributo>
                    <nombre>id_producto</nombre>
                    <tipo>numerico</tipo>
                    <formato></formato>
                    <estado>1</estado>
                </atributo>
                <atributo>
                    <nombre>nombre_producto</nombre>
                    <tipo>String</tipo>
                    <formato></formato>
                    <estado>1</estado>
                </atributo>
        </concepto>
</fuente>
```

Una responsabilidad de este tipo de agente es comunicar los datos extraídos al agente coordinador, para que este proceda a realizar el trabajo pertinente con los mismos. En caso de que se produzcan cambios en la fuente que un agente recolector monitorea, este debe comunicar tales cambios al agente coordinador y esperar respuesta del mismo, con la acción más pertinente a tomar. Por su parte, el agente almacenista se encarga de recibir los datos recopilados, almacenarlos, modificarlos y posteriormente enviarlos para su almacenamiento en la bodega de datos.

El agente integrador posee un conocimiento especializado de las estructuras de las diversas fuentes, el esquema unificado y las transformaciones requeridas para llevar los datos enviados por los agentes recolectores al esquema unificado. Este agente además debe identificar si los datos transformados pueden enviarse para ser almacenados en la bodega de datos, o deben mantenerse en una base temporal debido a diversos factores (procesos de análisis de datos sobre la bodega de datos, datos complementarios de la misma o de otras, etc.). Es de esta manera que la principal función

de este agente es transformar los datos, mantener actualizada la bodega de datos y también comunicar posibles eventualidades al agente interfaz, en caso de que el sistema no esté preparado para responder a cierto tipo de eventualidad (registros no trasladables al esquema unificado, etc.).

El agente coordinador está encargado de distribuir los agentes recolectores en las diferentes fuentes de interés y proporcionar el conocimiento tanto para acceder a ellas, como para extraer los datos requeridos. En caso de que las fuentes cambien, este agente es el encargado de comunicar al agente interfaz la eventualidad ocurrida. Cualquier acción a tomar sobre un cambio en una fuente, debe ser comunicada por el agente coordinador al agente encargado de la misma, para que continúe su trabajo.

El agente interfaz tiene la responsabilidad de enlazar el SMA en general con el usuario. Este debe brindar un canal de comunicación para mediar en las tareas que no son completamente automatizables (que por tanto requieren de alguna intervención), comunicar al usuario todo tipo de eventualidades y llevar al sistema las diferentes acciones que el usuario desee tomar ante éstas. También informa al usuario del estado de las fuentes de interés y muestra indicadores del comportamiento de las mismas. Este agente da acceso visual a los usuarios de los registros tanto de la bodega de datos como de la base temporal.

Finalmente, en cuanto al agente analista, en este caso no se trata de un agente de software sino de un humano que se encarga de velar por el buen desarrollo de las etapas de interés, realizando intervenciones únicamente cuando estas se encuentren por fuera de las capacidades de los demás agentes.

Como puede verse también en la Fig. 1, la solución propuesta involucra otros elementos, como son la bodega de datos, la base de datos temporal, el conocimiento de la bodega de datos y el conocimiento de las fuentes. En la primera se almacenan todos los registros capturados por los agentes recolectores y transformados por el agente integrador. Ésta es diseñada bajo un esquema unificado que busca lograr almacenar la diferente información de las diversas fuentes, teniendo en cuenta los procesos analíticos que se tienen como objetivo. La segunda es una base datos que permite al agente integrador almacenar datos temporalmente y ejercer trans-

formaciones sobre los mismos. Esta base asiste el almacenamiento temporal de datos que requieren ser complementados por otros para indexarlos como un solo registro a la bodega de datos. Como se pueden generar datos en las diversas fuentes continuamente, esta base se utiliza también como almacenamiento temporal de datos ya transformados, debido a que procesos posteriores, como por ejemplo minería de datos (Zhan & Zhan, 2003), visualización, respaldo, etc., pueden ser realizados sobre la bodega de datos en cualquier momento y no es conveniente realizar inserciones de datos con consultas en desarrollo.

En el conocimiento de la bodega de datos se define la base de reglas que son necesarias para el trabajo del agente integrador y que pueden almacenarse a través de reglas lógicas del tipo *SI X ENTONCES Y*. Estas reglas junto con metadatos buscan determinar una estructura adecuada para traducir los diferentes datos enviados de las fuentes al esquema integrador.

El conocimiento de la fuente es usado para que los agentes recolectores accedan y monitoreen las fuentes, de manera que se les permita identificar cuáles son los datos de interés, en qué tipo de sistema se encuentran (bases de datos, hojas de cálculo, internet, textos planos, etc.) y que métodos de identificación de cambios son requeridos, etc. Este conocimiento es transmitido a cada agente recolector cuando le es asignado una fuente por medio del agente coordinador y posee una organización específica dependiendo del tipo de fuente.

3. EXPERIMENTACIÓN Y RESULTADOS

Para llevar a cabo una validación del modelo propuesto, se empleó un caso de estudio controlado en el que se tomaron 3 fuentes de información de las cuales dos son construidas con datos simulados y una posee datos reales. El caso de estudio consiste en la información de una institución educativa, en este caso la Facultad de Minas de la Universidad Nacional de Colombia. Una primera fuente es un Sistema de Información Académica (simulada) en el cual se almacenan datos de tipo personal y académico de todos los estudiantes; una segunda fuente presenta documentos relaciona-

dos a Solicitudes de Facultad (simulada) como son las cancelaciones y aplazamientos de semestre; la tercera y última fuente es un Sistema de Gestión de Cursos (real), en este caso Moodle, con datos sobre cursos virtuales activos, participación y desempeño de estudiantes.

Para la validación del modelo se construyó un prototipo que busca recopilar e integrar la información disponible de estudiantes en los diversos sistemas. En la Fig. 2, que puede entenderse como una instanciación al caso de estudio de la Fig. 1, se puede apreciar la arquitectura del prototipo y las diversas tecnologías utilizadas.

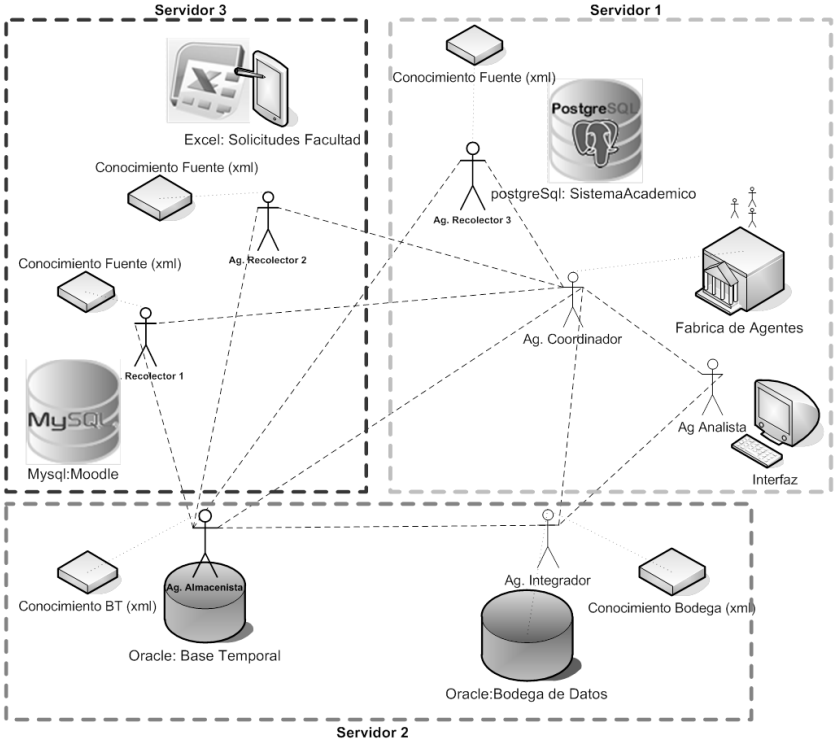

Fig. 2. Modelo de la organización del caso de estudio. Fuente: Autores

Los datos del caso de estudio están distribuidos físicamente en tres servidores cuyas características se presentan en la Tabla 1.

En el Servidor 1 se encuentran el sistema gestor de cursos Moodle y las Solicitudes de Facultad. En el Servidor 2 se encuentra el Sistema de Información Académica. Por el último en el Servidor 3 están alojadas la base temporal y la bodega de datos.

Tabla 1. Características de hardware de los servidores. Fuente: Autores

Servidor	Procesador	Memoria RAM	Disco Duro
1	Atlon XP 1.6 Ghz	1 Gb	250 Gb
2	Atlon X2 3 Ghz	2 Gb	500 Gb
3	Turion X2 ultra 2,2 Ghz	4 Gb	360 Gb

De acuerdo a la distribución física y al número de fuentes disponibles en el caso de estudio, se presenta un SMA distribuido en una LAN, donde se encuentran 3 agentes recolectores destinados cada uno a una fuente específica. Los Agentes Recolectores 1 y 2 se encuentran alojados en el Servidor 1 y el Agente Recolector 3 en el Servidor 2. Un Agente Almacenista encargado del manejo de la base temporal así como un Agente Integrador responsable de los procesos de integración y manejo de la bodega de datos se ubicaron en el Servidor 3. El Agente Coordinador encargado de la distribución y control de los agentes recolectores se alojó en el Servidor 2, mismo lugar donde el Agente Analista estuvo ubicado.

Con la finalidad de medir y verificar el adecuado funcionamiento del modelo propuesto se definieron los siguientes seis indicadores de resultados.

Indicador de Efectividad en la Recopilación (IER): Mide el resultado del acceso y captura de registros asociados a las distintas fuentes de interés, como se muestra en (1).

$$IER = \left(\frac{RC}{RTIF} \right) * 100\,[\%] \tag{1}$$

Donde, *RC*: registros capturados de las fuentes, y *RTIF*: registros totales de interés en las fuentes

Indicador de Efectividad en Carga Base Temporal (IECBT): Mide el resultado del proceso de carga de todos los registros recopilados en la base temporal, como se muestra en (2).

$$IECBT = \left(\frac{RABT}{RC} \right) * 100\,[\%] \tag{2}$$

Donde, *RABT*: registros almacenados en la base temporal.

Indicador de Efectividad en Carga Base Temporal según la fuente de procedencia (IECBF$_i$): Mide el resultado del proceso de carga de todos los registros recopilados en la base temporal, pertenecientes a una fuente *i*, como se muestra en (3). Este indicador es de interés, dado que la mayoría de problemas de carga de registros en base temporal se deben a la procedencia de registros de fuentes semi-estructuradas o no estructuradas.

$$IECBF_i = \left(\frac{RABTF_i}{RCF_i} \right) * 100\,[\%] \tag{3}$$

Donde, *RABTF$_i$*: registros almacenados en la base temporal procedentes de la fuente *i*, y *RCF$_i$*: registros capturados de la fuente *i*.

Indicador de desempeño del proceso de integración y carga (IDIC): Permite calcular eficiencia en la realización del proceso de transformación y carga de registros en la bodega de datos, como se muestra en (4).

$$IDIC = \left(1 - \frac{(errT + errC)}{errT + errC + RABD} \right) * 100\,[\%] \tag{4}$$

Donde, *errT*: número de errores cometidos en las transformaciones realizadas, *errC*: número de errores en la carga de registros a la bodega de datos, y *RABD*: registros almacenados en la bodega de datos.

Indicador de duración proceso de captura y carga de registros a la base temporal (IDCCT): Mide el tiempo invertido dado una arquitectura específica en desarrollar captura de datos de las fuentes y la carga de los mismos a la base temporal, como se muestra en (5).

$$IDCCT = Tcaptura + Tc\arg aBT\ [segs] \quad (5)$$

Donde, *Tcaptura*: duración total de la actividad de capturar todos los registros de las fuentes, y *TcargaBT*: duración total de la actividad de cargar todos los registros recopilados en la base temporal.

Indicador de duración proceso de integración (IDI): Mide el tiempo que se tarda en desarrollar las actividades de transformación y carga de registros a la bodega de datos, como se muestra en (6).

$$IDI = Ttransformacion + Tc\arg aBD\ [segs] \quad (6)$$

Donde, *Ttransformacion*: duración total de las operaciones de transformación de los registros que serán almacenados en la bodega de datos, y *TcargaBD*: duración total de la de registros a la bodega de datos.

Teniendo en cuenta los datos que se muestran en la Fig. 3, donde se compilan algunos resultados de interés de la ejecución del prototipo para el caso de estudio, se presentan a continuación los resultados para los indicadores definidos previamente.

Fig. 3. Resultados del prototipo en el caso de prueba. Fuente: Autores

Para el *Indicador de Efectividad en la Recopilación* se obtuvo, según (1), que *IER = (101956+858+8327)/111141 = 100%.*

El resultado arrojado por este indicador era de esperarse ya que las fuentes se instanciaron de forma local y no daban a lugar las restricciones de acceso, por tanto el único factor que podría afectar este indicador para el caso de estudio era una inadecuada definición del modelo del dominio, la cual no se presentó.

Para el *Indicador de Efectividad en Carga Base Temporal*, se obtuvo según (2) un valor de *IECBT = (101956+673+8327)/111141 = 99,8%.*

Según este indicador hubo una alta tasa de efectividad en la carga de datos a la base temporal, sin embargo denota que hubo un 0,02% de registros que no cumplieron con las restricciones de inserción presentadas en dicha base.

En el caso del *Indicador de Efectividad en Carga Base Temporal según la fuente de procedencia* se presenta el resultado para cada una de las tres fuentes según (3): $IECBF_1 = 101956/101956 = 100\%$, $IECBF_2 = 673/858 = 78,4\%$, $IECBF_3 = 8327/8327 = 100\%$

El resultado de las fuentes 1 y 3 es completamente satisfactorio, mientras que el de la fuente 2 (Solicitudes de Facultad) presenta un valor considerablemente más bajo. Esto se debe principalmente al sistema de almacenamiento en el que se encuentra el cual no posee restricciones de inserción y por tanto algunos registros pueden no cumplir las condiciones de las tablas en la base temporal. Esta situación no se presenta en las otras dos fuentes que son estructuradas por medio de un Sistema Gestor de Bases de Datos (SGBD).

Ahora, teniendo en cuenta que *errT = 2648*, *errC = 1072* y *RABD = 16096*, el *Indicador de desempeño del proceso de integración y carga* obtenido según (4) es *IDIC = 81,2%.* Este valor es un gran logro para esta investigación dada la complejidad de esta etapa. Los registros de integración resultantes en este caso de estudio son pocos pero esto es debido principalmente a la implementación de una regla que no permite llevar registros de estudiantes a la bodega si no se encuentran dentro el Sistema de Información Académica.

Para el quinto indicador (*Indicador de duración proceso de captura y carga de registros a la base temporal*) se procedió de la siguiente manera: Para calcular la duración total de la actividad de capturar todos los registros de las fuentes se tomó el valor inicial arrojado por el primer agente en comenzar a capturar registros y se halló la diferencia con el valor final del último agente en terminar de capturar sus respectivos registros. De esta forma, *Tcaptura = 11:27:31,165 - 11:27:32,266 = 1,101 s*. Luego, para calcular la duración total de la actividad de cargar todos los registros en la base temporal se realizó un procedimiento análogo tomando el valor inicial arrojado por el primer agente en comenzar a cargar registros y el valor final del último agente en terminar de capturar. Así, *TcargaBT = 11:27:31,271 - 11:27:58,365 = 27,094 s*. Una vez obtenidos estos valores, el resultado del indicador, según (5), es *IDCCT = 1,101 + 27,094 = 28,195 s.*

Finalmente, para calcular el sexto (*Indicador de duración proceso de integración)* se calcula primero el valor de la duración total de las operaciones de transformación aplicadas a los registros para su posterior almacenamiento en la Bodega. De esta manera, *Ttransformacion = 11:27:58,391 - 11:30:09,636 = 131,245 s*. Luego se calcula la duración de la actividad de cargar de registros a la Bodega: *TcargaBD = 11:30:09,636 - 11:30:13,556 = 3,920 s*. Con los que se obtiene que, según (6), *IDI = 131,245 + 3,920 = 135,165 s.*

Para el caso de los últimos dos indicadores, *IDCCT* e *IDI,* el análisis de la eficiencia de la propuesta sólo se podría hacer realizando una comparación contra otra alternativa de solución, esto es, realizando el proceso manualmente o, en el mejor de los casos, de manera asistida. Si bien no se cuentan con datos de lo que estas alternativas tardarían, si es posible decir que los presentados por la solución propuesta son aceptables (poco menos de 30 s en el primer caso y poco más de dos minutos y cuarto en el segundo).

4. CONCLUSIONES

Considerando la complejidad que involucran las actividades del proceso ETL, es clara la necesidad de desarrollar soluciones a través de enfoques innovadores que permitan incrementar la

automatización y eficiencia de procesos involucrados. Esto con la finalidad de reducir en la medida de lo posible el enorme esfuerzo, muchas veces manual, invertido por las organizaciones que deben realizar estas actividades como parte de sus procesos de negocio.

Con esto en mente, este artículo presentó una solución basada en agentes de software, y más específicamente en SMA, con el fin de aumentar la precisión de los datos obtenidos y mejorar así la calidad y velocidad de procesos posteriores de interés para tales organizaciones como pueden ser operaciones de minería de datos.

Para la construcción de tal solución se consideraron los problemas estructurales encontrados en aproximaciones conocidas (principalmente wrappers y soluciones ad-hoc) y se tuvo como norte lograr el mayor nivel de automatización posible en cada una de las tareas involucradas. En este sentido, más que una competencia, el modelo presentado es un complemento integrador de tales aproximaciones, tomando de cada uno sus principales fortalezas y logrando con esto generar una solución más completa a los diferentes problemas presentados en cada actividad del proceso ETL.

Precisamente, con los resultados obtenidos en un caso de estudio con tres fuentes de diversos tipos y situados de manera distribuida, se pudo determinar tanto la robustez como la eficiencia de la solución propuesta, siendo esto una motivación para su uso en otros casos de estudio. Este es precisamente parte del trabajo futuro que se tiene por delante, donde se buscarán aplicaciones en otros dominios tanto con datos simulados, como mixtos (como el mostrado en este artículo) y finalmente con datos completamente reales, que sean de interés para alguna organización particular. Cabe mencionar que justamente la arquitectura SMA presentada permite que tales aplicaciones en diversos dominios sean implementadas con relativa facilidad, simplemente definiendo el conocimiento de los agentes que se requieran incorporar.

Otro trabajo futuro es la validación de la propuesta ante escenarios donde se requiera procesamiento masivo de datos (cientos de miles o incluso millones de registros), aprovechando precisamente la naturaleza distribuida y paralela de los SMA.

5. REFERENCIAS

Boussaid, O., Bentayeb, F., Duffoux, A., Clerc, F. (2003); Complex Data Integration Based on a Multi-agent System, 201-212, Springer, Berlin.

Chawathe, S., Garcia-Molina, H., Hammer, J., Ireland, K., Papakonstantinou, Y., Ullman, J., Widom, J. (1994); The TSIMMIS Project: Integration of Heterogeneous Information Sources, Proceedings of IPSJ, Tokio, Japan, 7-18.

Di Fatta, G., Fortino, G. (2007); A Customizable Multi-Agent System for Distributed Data Mining, Proceedings of the ACM symposium on applied computing, 42-47, Seoul, Korea.

Ding, J., Guo, C-Z. (2009); Research of distributed ETL engine based on multi-agent and workflow, Journal of Computer Applications, 29(1), 319-322.

Franklin, S., Graesser, A. (1996); Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents, Third International Workshop on Agent Theories, Architectures, and Languages, 21-35, Budapest, Hungary.

Goasdoué, F., Lattes, V., Rousset, M. (2000); The use of CARIN language and algorithms for information integration: the PICSEL system, International Journal of Cooperative Information Systems, 9(4), 383–401.

Imtiaz, S., Hussain, A. (2005); Using Agents for Unification of Information Extraction and Data Mining, Proceedings of the International Conference on Information and Communication Technologies, 197-200, Karachi, Pakistan.

Jennings, N., Sycara, K., Wooldridge, M. (1998); A Roadmap of Agent Research and Development, Journal of Autonomous Agents and Multi-Agent Systems, 1(1), 7-38.

Rousset, M. C., Reynaud, C. (2004); Knowledge representation for information integration, Information Systems, 29(1), 3-22.

Simitsis, A., Vassiliadis, P., Sellis, T. (2005); Optimizing ETL Processes in Data Warehouses, Proceedings of the 21st International Conference on Data Engineering, 564-575, Tokyo, Japan.

Squire, C. (1995); Data Extraction and Transformation for the Data Warehouse, Proceedings of the ACM SIGMOD international conference on Management of data, 446-447, New York, USA.

Vassiliadis, P., Simitsis, A., Skiadopoulos, S. (2002); Conceptual Modeling for ETL Processes, Proceedings of the 5th ACM international workshop on Data Warehousing and OLAP, 14-21, Virginia, USA.

Viana, N., Raminhos, R., Moura-Pires, J. (2005); A Real Time Data Extraction, Transformation and Loading Solution for Semi-structured Text Files, Lecture Notes in Computer Science, 3808, 383-394.

Zhan, K., Zhan, C. (2003); Data preparation for data mining, Applied Artificial Intelligence, 17(5-6), 375–382.

Zhang, J., Ghen, H-G. (2010); Research on real-time ETL system model based on multi-agent, Information Technology, 2010(2), 71-73.

Zhou, G., Hull, R., King, R. (1996); Generating Data Integration Mediators that Use Materialization, Journal of Intelligent Information Systems, 6(2-3), 199-221.