Desarrollo de un Agente Inteligente Basado en el Estándar ANSI/ISA-95

Melina Vidoni[1], Aldo Vecchietti[2]
Instituto de Desarrollo y Diseño, Ingar CONICET-UTN - Santa Fe, Argentina
[1]melinavidoni@santafe-conicet.gov.ar, [2]aldovec@santafe-conicet.gov.ar

Resumen: *Los cambios en las organizaciones y en la integración interorganizacional ha generado una necesidad de estandarizar las estructuras de datos empleadas para aumentar la eficiencia del flujo de información. El estándar ANSI/ISA-95 es relevante como medio para la estandarización y automatización de sistemas empresariales en empresas de producción junto con la estructura de información de manufactura que define. Este trabajo propone a GrACED, un agente inteligente basado en conocimiento que procesa lenguaje natural mediante bolsas de palabras, para analizar y clasificar la estructura de las tablas de las base de datos de los ERP en las categorías propuestas por el ANSI/ISA-95. El objetivo es promover un medio adaptable, portable y de base estandarizada para analizar automáticamente la información contenida en cada tabla de la base de datos y estudiar la adecuación de dicho ERP al S95 buscando facilitar el estudio del sistema empresarial y favoreciendo su integración con otros sistemas.*

Palabras Claves: *ERP, agente inteligente, ANSI/ISA-95, bases de datos, información de manufactura.*

INTRODUCCIÓN

Un importante desafío para las organizaciones es el cambio de sus entornos, lo que implica una alta necesidad de flexibilidad, agilidad, eficiencia y calidad en sus procesos. Debido a esto la Comisión Europea (EU-Commission, 2004) recomendó la mejora de los procesos de integración ínter-organizacionales y en la cadena de suministros a través de su estandarización y posterior automatización. La toma de decisiones integradas y la optimización colaborativa dentro de las empresas pasó a tener un rol crucial en la interrelación de organizaciones. Con este objetivo en mente muchas han desarrollado sistemas tipo MES (Manufacturing Execution Systems) o CPM (Collaborative Production Management) con una única finalidad: anular la brecha entre los procesos, las comunicaciones y los sistemas ERP (Enterprise Resource Planning; Harjunkoski, Nyström y Horch 2009).

Para alcanzar esta integración es imperativo definir estructuras de información y herramientas sofisticadas que permitan explotar dichas configuraciones con el objetivo de mejorar la disponibilidad y comunicación de los datos, más específicamente de manufactura, si lo que se desea es integrar cadenas de suministro. Siguiendo esta línea se han propuesto muchos estándares para mejorar la eficiencia y el flujo de la información de manufactura, entre ellos el ANSI/ISA-95 (Muñoz y otros, 2012).

ANSI/ISA-95 es un estándar internacional para desarrollar interfaces automatizadas entre empresas y sistemas de control que propone un conjunto de modelos y definiciones fundadas en una terminología consistente para describir las tareas e información de manufactura y producción que deben ser intercambiadas en sistemas que se interrelacionan (ISA, 2000). En los últimos años este estándar ha sido ampliamente aceptado debido a que especifica un modelo funcional completo (Prades y otros, 2013).

Se han realizado varios trabajos académicos para favorecer el intercambio de información estandari-

zada según los modelos del ANSI/ISA-95, así como diferentes formas de implementación. En 2009 (Harjunkoski, Nyström y Horch, 2009) Har junkoski y otros propusieron una plataforma para el intercambio de información utilizando diagramas BPMN (Business Process Model and Notation) basados en los modelos del ANSI/ISA-95. También se han efectuado avances en el área de simulación con el objetivo de generar especificaciones para desarrollar sistemas uniformes (Kardos y otros, 2013). Otros autores (Muñoz y otros, 2012) emplean ontologías para generar un framework que integra la toma de decisiones utilizando las estructuras del ANSI/ISA-88. Finalmente, He y colab. (He, Lobov y Matínez Lastra, 2012) generaron una herramienta para el modelado de empresas con fundamentos en el ISA-95 y en el IEC 62246.

Sin embargo, si bien muchos estudios se enfocan en diseñar nuevos sistemas y herramientas basados en el ANSI/ISA-95 (Brandl, 2002), muy pocos intentan analizar los ya existentes y proveer un informe sobre su adecuación al estándar, o estudiar qué tipo de información de manufactura contienen a la luz de la clasificación propuesta en el ANSI/ISA-95. Este análisis favorecería la integración entre sistemas sin obligar a las empresas a cambiar radicalmente su forma de trabajo. También permitiría proveer un marco para el análisis de los mismos y su modificación con el objetivo de adecuarse al ANSI/ISA-95.

En un desarrollo previo (Vidoni y Vecchietti, 2013), se propone un prototipo de un sistema tipo APS (Advanced Planning and Scheduling) y se establece la necesidad de presentar al usuario alguna indicación sobre la ubicación de la información de manufactura en la base de datos del ERP, con el objetivo de enlazar los modelos matemáticos del APS con el ERP empleado.

Utilizando esta idea como disparador inicial, la propuesta que se describe en este artículo es utilizar un agente inteligente, cuya base de conocimiento esté dada por el ANSI/ISA-95 y que pueda clasificar el contenido de la base de datos de un ERP o de un sistema empresarial, en cada una de las categorías que el estándar propone empleando el enfoque de bolsas de palabras. Una de las fortalezas de utilizar un agente inteligente es la capacidad inherente del mismo de procesar el lenguaje natural.

En esta línea también se han realizado proyectos sobre categorización de textos o estructuras utilizando agentes inteligentes. Uno de estos trabajos (Quiñonez Gámez y Camacho Velázquez, 2011) propone una clasificación sobre fuentes de generación de gas utilizando algoritmos genéticos y redes neuronales para posteriormente compararlos. Otros autores (Fu, Ke y Mostafa, 2005) formulan un método de clasificar documentos de texto de forma automatizada usando un conglomerado de múltiples agentes que procesaban lenguaje natural; en este enfoque cada agente sólo catalogaba en una sola categoría. Finalmente la idea de bolsas de palabras también se emplea, a través de un modelo bayesiano, para la generación de documentos de texto usando agentes inteligentes (Wallach, 2006).

Cabe destacar que hasta el momento no se han encontrado trabajos que empleen agentes inteligentes para analizar sistemas existentes a la luz de los conceptos propuestos por el estándar ANSI/ISA-95.

IMPLEMENTACIÓN DE LA BASE DE CONOCIMIENTO

Siguiendo la definición de Russel y Norvig (Norvig y Russel, 2010), un agente inteligente es una entidad autónoma inserta en un ambiente que advierte lo que sucede en él a través de percepciones (realizadas mediante sensores) y responde a ellas

actuando de forma racional a través de acciones ejecutadas por actuadores. Más específicamente, los agentes tipo knowledge based (basados en conocimiento) especializan la definición anterior y poseen una representación del conocimiento y un proceso de razonamiento que lo ejecuta y puede combinarlo con las percepciones para poder inferir aspectos ocultos del estado actual antes de seleccionar acciones. Estos agentes son utilizados para procesar lenguaje natural dado que su comprensión radica en inferir los estados ocultos, es decir, la semántica detrás de las palabras.

Respecto a los ERP, la persistencia de la información de estos sistemas se realiza en sus bases de datos, las cuales son en su mayoría de tipo relacional. De esta forma, si se quiere analizar cómo se organiza la información en un ERP, es necesario analizar y clasificar la estructuración de los datos en las tablas de su base de datos.

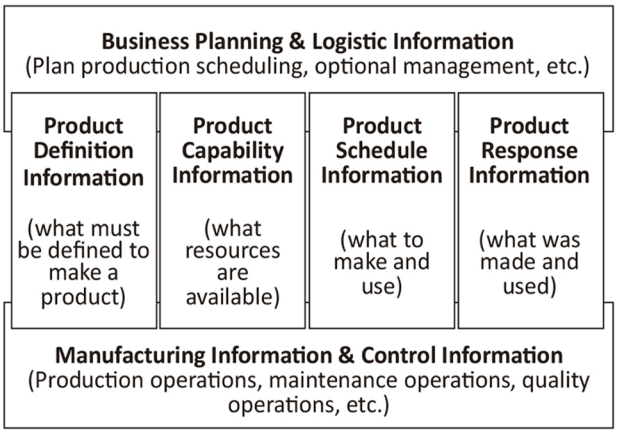

Figura 1 - Categorías de información de ANSI/ISA-95.

Para poder clasificar es imperativo tener categorías definidas y estandarizadas que tengan una aceptación moderada a amplia y que formen parte de la base de conocimiento (BC) del agente inteligente.

Esto mismo fue lo que llevó a la utilización y aplicación del ANSI/ISA-95, un estándar internacional también conocido como S95 o ISA95, que propone modelos y definiciones consistentes para generalizar la estructura y nombramiento de las tareas e información de manufactura y producción (ISA, 2000). Más específicamente en la Parte 3 (ISA, 2005) clasifica la información de manufactura en cuatro categorías que definen la información de productos y de producción, las cuales pueden observarse en la Figura 1.

De estas categorías se decidió trabajar sólo con Product Definition, Production Capability y Production Schedule. Esta decisión fue tomada debido a que la información de la categoría Production Response se representa, a menudo, sólo en atributos de las tablas y no en tablas completas, lo que aumenta la complejidad de la clasificación.

Por otro lado la Parte I del estándar (ISA, 2000) contiene las definiciones y conceptos que son posteriormente utilizados para generar la estructura y definir las categorías en las que se va a clasificar. Como parte de estas definiciones el estándar propone gráficos de superposición que explicitan subcategorías de información para cada categoría de la Figura 1, definen las subcategorías y cómo se superponen entre ellas.

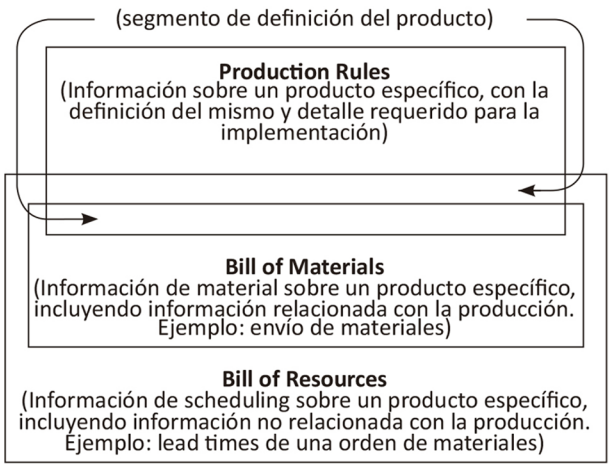

Figura 2 - Superposición de información en la Definición del Producto, para ISA-95 Parte I.

En la Figura 2 se observa el gráfico de superposición para la categoría Product Definition que es la que se ha utiliza, a posteriori, en los casos de estudio con el objetivo de acotar las primeras evaluaciones del agente.

GENERACIÓN DE CATEGORÍAS

De esta forma, para poder generar la base de conocimiento, se estructuraron las categorías y subcategorías de información de manufactura presentadas en el ANSI/ISA-95 en un grafo, el cual puede observarse en la Figura 3. Los nodos ovalados representan las categorías que pueden utilizarse para clasificar mientras que los rectangulares de bordes redondeados son presentados en el estándar pero no se van a emplear en la clasificación.

El nodo raíz representa a la totalidad de información de manufactura, mientras que los nodos de nivel 1 son las grandes categorías de la Figura 1 (sin usar a Production Response, como se mencionó previamente). Por otro lado los nodos de niveles sucesivos e inferiores fueron obtenidos de los gráficos de superposición (por ejemplo las categorías visualizadas en la Figura 2 son representadas como los hijos del nodo Product Definition en el grafo de la Figura 3 y de las descripciones de cada categoría).

BOLSAS DE PALABRAS

En el siguiente paso para generar la base de conocimiento se propone un método que permita emparejar las tablas de la base de datos del ERP con una o más categorías de las presentadas en el grafo. Más allá del Sistema de Gestión de Base de Datos (SGBD) que se emplee, tanto las tablas como las columnas tienen nombres que las identifican, los cuales se eligen para darle significado semántico al contenido que almacenan. Por esto mismo se decide trabajar con lenguaje natural clasificando las tablas por las palabras que la definen.

El enfoque utilizado es el de bolsas de palabras (o bag of words, abreviado BoW). Esto es una representación simplificada que se usa en el procesamiento de lenguaje natural donde cada clase o documento se representa en un conjunto múltiple (o bolsa) de palabras sin considerar la gramática (formación de sentencias) ni el orden de las palabras (Wallach, 2006).

Se debe mencionar que el estándar sólo define

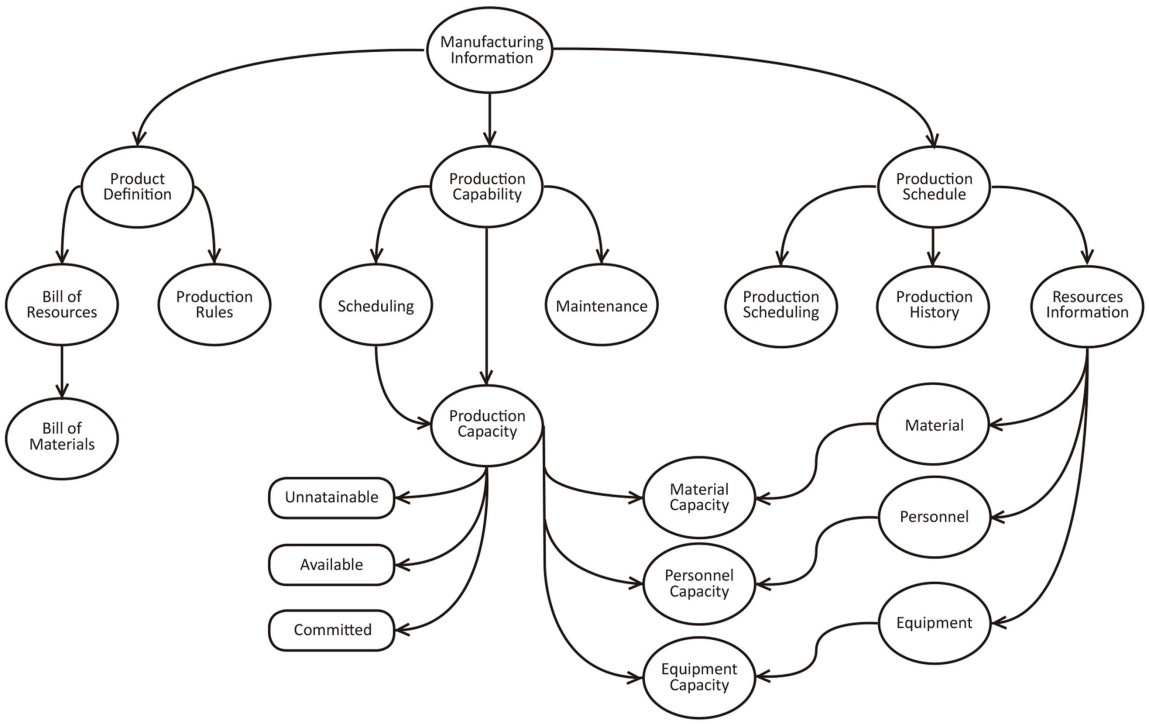

Figura 3 - Grafo de Categorías derivado del ANSI/ISA-95 Parte I.

las categorías indicando qué tipo de información se incluye pero sin proporcionar las palabras para formar las bolsas. La generación de éstas fue parte del desarrollo de este proyecto y se expone en la siguiente sección.

Sin embargo no todas las palabras tienen la misma relevancia, la cual incluso puede variar de categoría en categoría; por esto mismo se decidió asignar pesos a las mismas. Cada bolsa tiene un peso total de 100, el cual fue dividido internamente entre las palabras, dado un peso mayor a las que son más representativas. Además estos pesos dependen de la categoría que representan.

Un punto importante para mencionar es que, muy a menudo, las palabras que se utilizan para definir nombres de tablas no suelen ser las mismas que se emplean en los nombres de las columnas, aún cuando pertenezcan a la misma categoría. Debido a ésto se decide asociar dos bolsas de palabras por categoría (o nodo ovalado en el grafo de la Figura 3): una para las palabras en los nombres de las tablas y otra para las columnas.

Otro aspecto que se debe considerar es el uso de sinónimos, palabras escritas de diferente forma pero que significan lo mismo o abreviaturas, convenciones ortográficas que acortan la escritura de cierto término o expresión. Agregar cada combinación para cada palabra a las bolsas no es conveniente ya que no sólo introduce redundancia y aumenta el tiempo de procesamiento, sino que además reduce los pesos de las palabras dentro de las bolsas; esta redundancia también impacta en el porcentaje de pertenencia final de una tabla a una categoría.

Como consecuencia de la riqueza de los lenguajes naturales, las palabras pueden tener distintos significados, sinónimos y abreviaturas. Estas variaciones dependen de la categoría en la que está siendo clasificada la palabra (por ejemplo, tanto product como production pueden ser abreviadas como prod). Por esto se agregaron archivos exclusivos de sinónimos y abreviaturas que fueron relacionados directamente a cada palabra en cada bolsa. Estos archivos son directamente nombrados como Archivo de Sinónimos, incluso si contienen abreviaturas y acrónimos.

IMPLEMENTACIÓN DE LA BASE DE CONOCIMIENTO

Como se explicó anteriormente se asociaron dos BoW por cada nodo del grafo de la Figura 3. Por este motivo se decide almacenarlo como el índice de la base de conocimiento del agente que contiene las referencias a las categorías y a las bolsas de palabras pero manteniendo las relaciones de niveles. Esta implementación se realizó utilizando archivos XML (W3C Recommendation, 2006) y siguiendo la estructura presentada en la Figura 4.

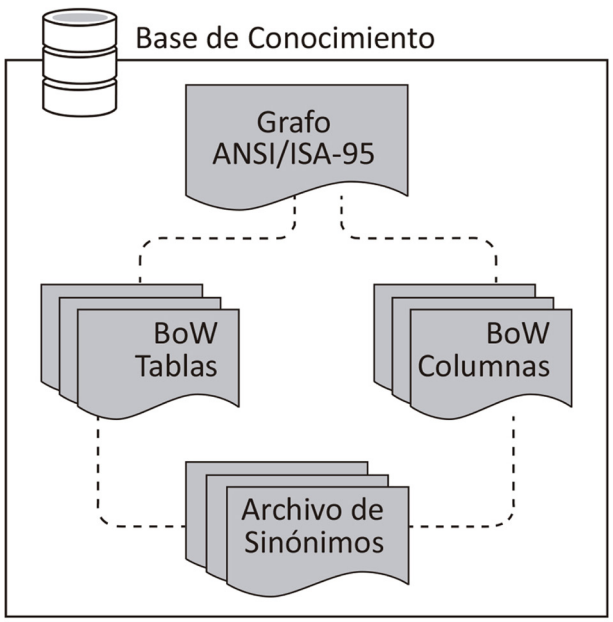

Figura 4 - Estructura de la base de conocimiento propuesta, basada en el ANSI/ISA-95.

```
<! - -BILL OF MATERIALS NODE- - >
<tns:node  tns:nodeName=”Bill Of Materials”  tns:columnameBow=”bom_col.xmL”
tns:tablenameBow=”bom_tab.xmL”  tns:usable=”true”>
    <tns:relation  tns:relationName=”part of” />
</ tns:node>
```

Figura 5 - Extracto de código XML para el grafo ANSI/ISA-95basada en el ANSI/ISA-95.

De este modo, debido a que el grafo se guarda como un archivo XML, cada nodo es un elemento dentro del mismo, el cual tiene atributos para el nombre y para el nombre de archivo de cada bolsa de palabra diferenciando el uso de cada una. En la Figura 5 puede observarse un extracto del código XML que representa al nodo Bill Of Materials donde los atributos columnameBow y tablenameBow son los que guardan el nombre de archivo de las respectivas bolsas de palabras.

El procedimiento que se emplea para generar las bolsas de palabras y los archivos de sinónimos es manual y se describe a continuación:

1. Se listan los nombres de tablas y las columnas de cada ERP.

2. Para cada ERP:

a. Se clasifica manualmente cada tabla considerando las descripciones de contenido del estándar ANSI/ISA-95.

b. Se separan las palabras que conforman cada nombre de tabla y de columnas. Por ejemplo el nombre de tabla stock_inventory_move se transformó en tres palabras: stock, inventory y move.

3. Manteniendo la distinción del origen de las palabras (es decir, si eran de los nombres de tablas o de los nombres de columnas) se agrupan todas las palabras de cada categoría (todas las pertenecientes a Bill of Materials, las pertenecientes a Production Rules, etc.).

4. Para cada grupo de palabras:

a. Se cuenta la cantidad de veces que aparece cada palabra para obtener la "relevancia" o "nivel de descripción" que aporta la misma para una categoría.

b. Separadamente se anota cada palabra y los sinónimos de la misma.

c. Se suma la cantidad de apariciones de la palabra y sus sinónimos.

d. Dándole un peso total de 100 a cada bolsa de palabras se otorga un peso a cada palabra considerando la cantidad de apariciones encontradas en el punto anterior.

DESARROLLO DEL AGENTE INTELIGENTE

Basándose en la definición de Russel y Norvig (Norvig y Russel, 2010) se propone un agente inteligente denominado GrACED (por las siglas en inglés de Grammar Agent for Classifying ERP Databases: Agente Gramático para Clasificar Bases de Datos de ERPs). Siguiendo los componentes mencionados en dicha definición, en la Figura 6 puede observarse la estructura básica de la propuesta.

De este modo GrACED se inserta en un ambiente, el cual está representado por el ERP que se desea analizar. Este ambiente posee un estado compuesto por los datos de conexión a la base de datos del ERP y la lista de las tablas existentes en dicha base, que son las que se van a analizar y clasificar.

Por otro lado GrACED tiene dos percepciones relacionadas entre sí: obtener el nombre de tabla a analizar y luego los nombres de las columnas de dicha tabla. Estas percepciones son almacenadas en el estado del agente mientras se está ejecutando la única acción que posee: Clasificar. Los otros componentes del estado del agente lo enlazan a la base de conocimiento (la cual se explica en la siguiente sección) y a una lista temporal de las preclasificaciones obtenidas para la tabla que está analizando mientras la acción Clasificar se está ejecutando.

Finalmente el agente también tiene una prueba de meta, la cual le permite evaluar si ha llegado a su objetivo o si aún necesita continuar trabajando.

ALGORITMO DE RAZONAMIENTO

El algoritmo de razonamiento es ejecutado durante la acción de Clasificar con el objetivo de emparejar cada tabla con una o más categorías que representen la información que contiene. Para esto emplea los nodos "habilitados" del grafo y que sirve para encontrar qué tipo de información almacena cada tabla.

Dado que hay dos bolsas de palabras por categoría, este algoritmo también tiene dos pasos: el primero es tomar el nombre de la tabla y clasificarlo utilizando las bolsas de palabras que tiene para ese efecto. Este proceso puede verse en la Figura 7.

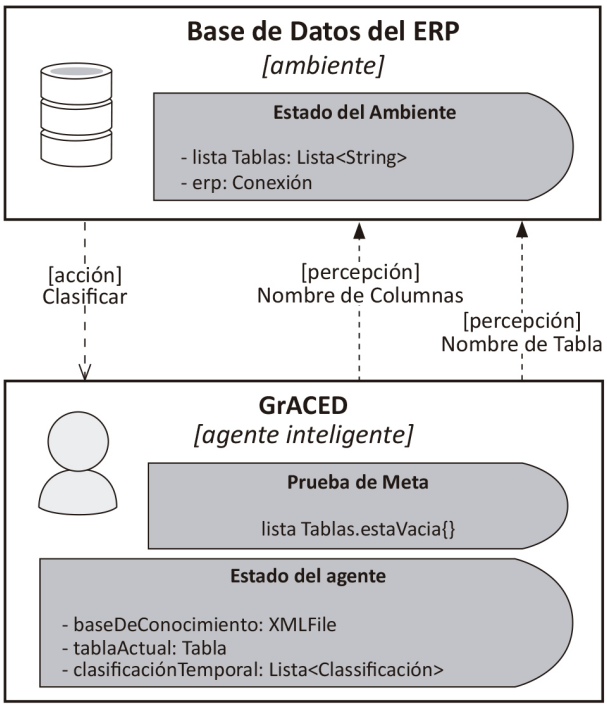

Figura 6 - Estructura básica del agente inteligente.

Puede notarse que hay dos “filtros” para determinar si una clasificación es adecuada o no. El primero de ellos se realiza comparando la cantidad total de palabras del nombre de una tabla contra la cantidad de ellas que fueron encontradas en la BoW; para poder pasar a la siguiente etapa al menos la mitad de las palabras del nombre debe estar en la bolsa. El segundo filtro implica calcular el porcentaje de pertenencia (sumar los pesos de las palabras encontradas) y si dicho valor es menor a un 10%, la clasificación se descarta.

El porcentaje del segundo filtro se selecciona siguiendo las premisas detalladas a continuación. En promedio el nombre de una tabla no suele tener más de cuatro palabras, mientras que cada bolsa de palabra para los nombres de tablas debe guardar entre 25 y 30 registros ya que hay una amplia variedad que puede usarse para cada categoría. Esto hace que encontrar más del 10% de las palabras de la bolsa en el nombre de la tabla genere una clasificación de importancia. Por esto mismo más adelante se detallan las tipificaciones que GrACED realiza con las clasificaciones obtenidas.

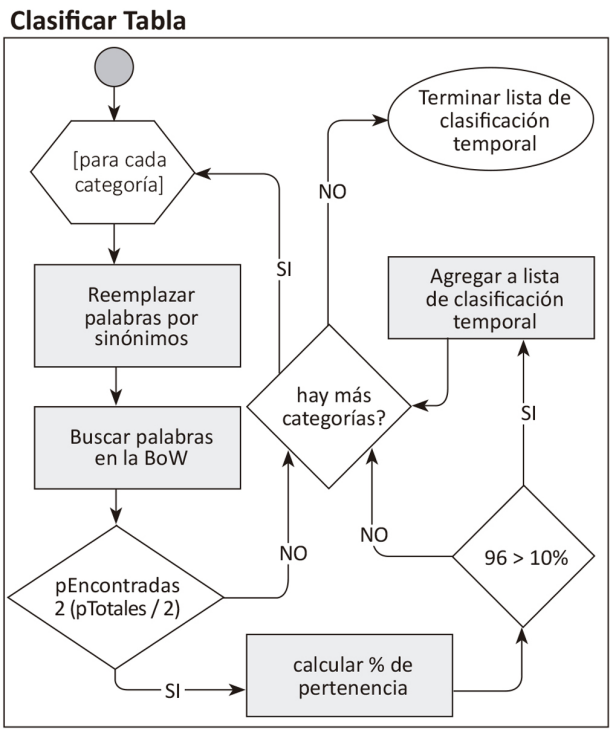

Figura 7 - Preclasificación de tablas - Parte 1 del Algoritmo de Razonamiento.

En el segundo paso de la clasificación se evalúan las palabras en los nombres de las columnas, sólo en las categorías que sobrepasaron la clasificación con el nombre de la tabla. Los pasos que se efectúan son muy similares a los de la Figura 7 pero con un sólo filtro que es evaluar que el porcentaje de pertenencia obtenido es mayor que un 15%. Si el filtro no es superado la categoría es removida de la preclasificación de la tabla.

PERSISTENCIA DE LOS RESULTADOS

La acción del agente inteligente finaliza al persistir los datos de las clasificaciones en dos archivos XML:

- Tablas Clasificadas: contiene las que han sido clasificadas en al menos una categoría, el nombre de la categoría, y los porcentajes de pertenencia.

- Tablas No Clasificadas: contiene las que no pertenecen a una clasificación. Este segundo archivo existe debido a que no todas las tablas de una base de datos de un ERP contienen información de manufactura.

Por otro lado GrACED también ofrece gráficos en su interfaz de usuario al finalizar la clasificación, lo que le permite al usuario poder realizar un estudio más complejo de los resultados obtenidos. Dichos gráficos son:

a. Un gráfico de torta con la proporción de tablas que contienen información de manufactura y las que no. Esto es especialmente útil para analizar la distribución de los datos y la relevancia que cada empresa le da a los mismos.

b. Un gráfico de barras para cada tabla mostrando las categorías en las que fue clasificada y el porcentaje que obtuvo al ser analizada por nombre de tabla, por nombre de columna y el promedio. Es decir que cada tabla puede pertenecer a más de una categoría debido a la ambigüedad del lenguaje natural y a la superposición de los datos.

c. El último gráfico es un diagrama de torta que asigna un tipo a cada uno de los promedios de clasificación con el objetivo de obtener mayor información respecto a las clasificaciones. Este tema es desarrollado a continuación con mayor profundidad.

Dado que por cada categoría hay dos bolsas de palabras disyuntas (una para tablas y otra para columnas), cada una genera un porcentaje de pertenencia distinto. El motivo para esto es que se puede generar un análisis mucho más valioso al evaluar dónde se presenta una pertenencia más fuerte, ya sea en el nombre de la tabla o en el de las columnas.

Estos dos valores son tipificados en uno de los siguientes tipos. A su vez figuran en el gráfico detallado anteriormente en el ítem c).

- Falsos Positivos (Tricky): son clasificaciones en las que la pertenencia del nombre de la tabla es mucho mayor que la obtenida con los nombres de las columnas. Esto sucede en casos donde el nombre de la tabla tiene palabras muy específicas para una categoría mientras que las columnas tienen palabras genéricas cuyos pesos son medios o bajos. Sin embargo no se descartan porque más allá de la combinación de pertenencias obtenidas la tabla puede contener información relevante.

SEPARACIÓN DE PALABRAS

Como puede notarse la implementación de un agente que procesa lenguaje natural depende de dos situaciones ajenas al mismo: la sintaxis y la semántica de las palabras. Si las palabras son escritas en idiomas que el agente no comprende, o con errores ortográficos, éste no podrá procesarlas. Lo mismo sucede si la semántica de las palabras no es utilizada adecuadamente; por ejemplo si una columna se llama cellphone_number pero en realidad contiene un nombre de persona física, el agente generará una clasificación basada en el nombre de la columna y no en el contenido de la misma ya que la semántica de la etiqueta ha sido usada erróneamente.

Un punto importante relacionado con las palabras es la separación de las mismas. Generalmente en los lenguajes de programación se utilizan convenciones de nombres (o naming conventions por el nombre en inglés) que establecen métodos para separar las palabras. Las bases de datos actuales no tienen ninguna convención preestablecida y, aunque la tuvieran, no hay forma de asegurar que los desarrolladores las utilizarían, por lo que el problema de la

separación no resulta trivial (Butler y otros, 2009).

Para solucionar ésto, antes de comenzar la clasificación, el agente solicita que se le instruya qué método de separación va a emplearse. Actualmente los tipos de separación comprendidos son:

- Pascal Casing: las palabras se escriben juntas y cada una empieza con letra capitalizada. Ejemplo: UnEjemploDePalabras.

- CamelCasing: similar al anterior, la primera palabra lleva letra minúscula. Ejemplo: unEjemploDePalabras.

- Separación por Caracteres: las palabras son escritas en minúsculas y cada una se separa usando un carácter especial (punto, espacio, guion medio, guion bajo). Ejemplo: un_ejemplo_de_palabras.

- Separación Mixta: es una separación más compleja y personalizable y permite seleccionar un prefijo que será eliminado y no analizado, una separación para el prefijo del resto del nombre y una para el nombre restante.

Cabe mencionar que si una base de datos no mantiene una semántica adecuada, ni consistencia en el método de separación de palabras, GrACED no clasificará a su máxima capacidad. Esto se muestra en uno de los casos que se analizan en la siguiente sección.

IMPLEMENTACIÓN Y CASOS DE ESTUDIO

La implementación de un agente inteligente es una tarea compleja por lo que se decidió utilizar FAIA (Roa y otros, 2010): un framework generado en Java que ofrece una estructura de clases abstractas que generan varios tipos de agentes inteligentes (reactivos, basados en metas, basados en conocimiento, etc.) y que sirven de marco para implementar la funcionalidad básica de todo agente (la entidad, el ambiente, estado del ambiente, estado del agente, percepciones y acciones).

A su vez, la base de conocimiento se implementa en XML debido a la portabilidad, flexibilidad y universalidad que este lenguaje ofrece, además de permitir una fácil modificación y agregado de palabras. Otra ventaja es que a partir de la versión Java 8 las librerías para la lectura/escritura de este tipo de archivos ya se encuentran incorporadas en el lenguaje quitando la necesidad de utilizar archivos JAR externos.

Con el objetivo de realizar una primera implementación y evaluar el comportamiento y la arquitectura propuesta para GrACED se trabaja inicialmente sólo con una rama del grafo de categorías (Figura 3). De este modo, en la Figura 8 puede observarse el grafo reducido sobre el cual se hace la implementación y la evaluación, donde los nodos con fondo sombreado son aquellos que se encuentran habilitados y poseen bolsas de palabras.

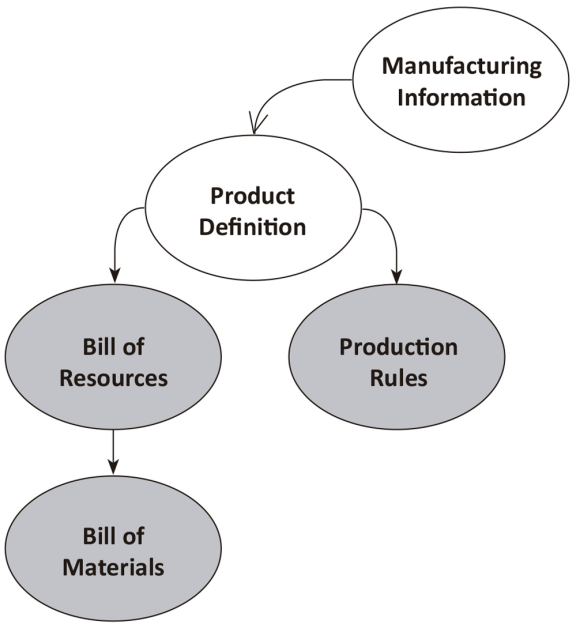

Figura 8 - Implementación inicial del grafo derivado del estándar ANSI/ISA-95.

Para generar las bolsas de palabras, los archivos de sinónimos y estudiar el comportamiento de GrACED se emplea cross-validation (Kohavi, 1995). Esta es una técnica para evaluar cómo los resultados de un análisis estadístico se generalizan en un conjunto independiente. Para esto, dado un set

de datos disponibles, se lo separa en tres partes: los datos del entrenamiento, los de la validación y los desconocidos (o de primera vista). Se puede repetir el seccionamiento de los datos y el posterior entrenamiento/evaluación con distintos subconjuntos con el objetivo de encontrar cual es el particionamiento de datos que genera el mejor entrenamiento posible.

De esta forma, de todos los ERP de código abierto seleccionados para la utilización, sólo se emplean cuatro para generar la base de conocimiento del agente (esto incluye BoW de tablas/ columnas y archivos de sinónimos): Compiere (Pretorious, 2010), OpenERP (OpenERP S.A., 2012), ERPNext (Panorama Consulting Solutions, 2010) y JFire (NightLabs Consulting GmbH, 2011). Se reservaron dos ERPs como conjunto de evaluación: Dolibarr (Destailleur, 2014) y Libertya (Cristina y otros, 2011) (este último no llegó a ser evaluado para este artículo). La modificación que se hizo fue agregar dos casos de estudio extra: OpenERP (ya utilizado en el subconjunto de entrenamiento) y Adempiere (Pamungkas, 2009).

A su vez el lenguaje natural para realizar la base de conocimiento es el idioma inglés. Cualquier palabra en otro idioma se trata como sinónimo de su correspondiente palabra en inglés.

Del grafo de la Figura 8, sólo los nodos Bill of Materials, Bill of Resources y Production Rules se emplean para clasificar y elaborar la base de conocimiento del agente. Los siguientes ítems detallan el tamaño final de la BC implementada en este trabajo:

- Cantidad de BoW para nombres de tablas: 3.
- Cantidad de BoW para nombres de columnas: 3.
- Cantidad de archivos de sinónimos: 189.
- Palabras en la bolsa para nombres de tablas: 70.
- Palabras en la bolsa para nombres de columna: 423.
- Proporción (columnnames per tablenames): 6,043.
- Palabras totales en las bolsas: 493.

OPENERP

OpenERP (OpenERP S.A., 2012) es una suite ERP de código abierto publicado con una licencia AGPL2 (GNU Affero, 2007) e implementado como una aplicación web. Su funcionamiento se centra en la lógica de negocios y en el módulo MRP. Esta suite también fue utilizada por los autores en el caso de estudio de una investigación previa (Vidoni y Vecchietti, 2013).

La base de datos de OpenERP fue implementada en PostgreSQL y mantiene consistencia en la convención de nombres usando siempre las letras en minúsculas separadas con guiones bajos. De esta forma m_production_id fue considerado como un nombre adecuadamente separado (o preciso), mientras que movementdate se consideró incorrecto por la falta de guion bajo entre ambas palabras.

Las estadísticas de precisión de la separación de palabras de la BD de OpenERP son:

- Total de nombres de tablas: 450.
- Nombres de tablas correctos: 416.
- Total de nombres de columnas: 4753.
- Nombres de columnas correctos: 4652.

Con estos datos se obtiene que el 92.44% de los nombres de las tablas de la BD están adecuadamente separados mientras que el 97.87% de los nombres de columnas tiene una separación correcta. Esto da un total de 97.405% de precisión en la base de datos.

Este ejemplo fue analizado con GrACED y algunos de los resultados pueden verse en la Figura 9 (arriba), donde se puede observar una de las pestañas con resultados generados por el agente donde del total de tablas el 8.89% contiene información sobre la categoría Product Definition y el 91.11% no.

Por otro lado, en la Figura 9 (abajo) se observa la separación en tipos de clasificación para todas las tablas que han sido categorizadas. Aquí se cuenta el total de clasificaciones ya que una tabla puede

pertenecer a más de una categoría; de esta forma hay una mayoría de categorizaciones de tipo neutral (77.36% del total) dado que en la mayoría de las etiquetas o nombres de tablas no se emplean palabras realmente representativas. Del total de categorizaciones un 13.21% fue tipificada como Positivo Total (pertenencia de nombre de tabla y de nombre de columnas mayor al 50%) y el restante 9.43% se consideró Falso Positivo.

Para evaluar el comportamiento obtenido con GrACED se compararon los resultados automatizados del agente contra una clasificación manual realizada por expertos. De esta forma los expertos realizaron 44 categorizaciones y GrACED coincidió con 40, lo que representa un 90.91% de certeza. A su vez el agente agregó 13 categorizaciones de las cuales 9 fueron posteriormente consideradas correctas por los expertos tras estudiar el contenido de información y palabras de las mismas.

DOLIBARR

Dolibarr (Destailleur, 2014) es un ERP de código abierto publicado bajo una licencia GNU General Public License 3.0 (Free Software Foundation Inc., 2007) orientado a empresas y compañías de tamaño medio. De origen francés, Dolibarr tiene más de 26 módulos, considerando entre ellos un catálogo de productos y servicios, administración de órdenes de venta y producción, envíos, entre otros.

Para este estudio se empleó la versión estable 3.5.2 liberada en Abril de 2014 y la base de datos fue implementada en MySQL. Similarmente a Open

ERP el método principalmente empleado para la separación de las palabras en dicha BD es el carácter especial guion bajo. Sin embargo este ERP no posee la misma precisión que en el caso de estudio anterior y esto puede verse a continuación:

- Total de nombres de tablas: 176
- Nombres de tablas correctos: 130
- Total de nombres de columnas: 1967
- Nombres de columnas correctos: 1855

Con estos datos Dolibarr posee un 73.86% de precisión en la separación de palabras de los nombres de tablas y un 94.31% de precisión en los nombres de columnas. Esto da una precisión promedio de 92.63%.

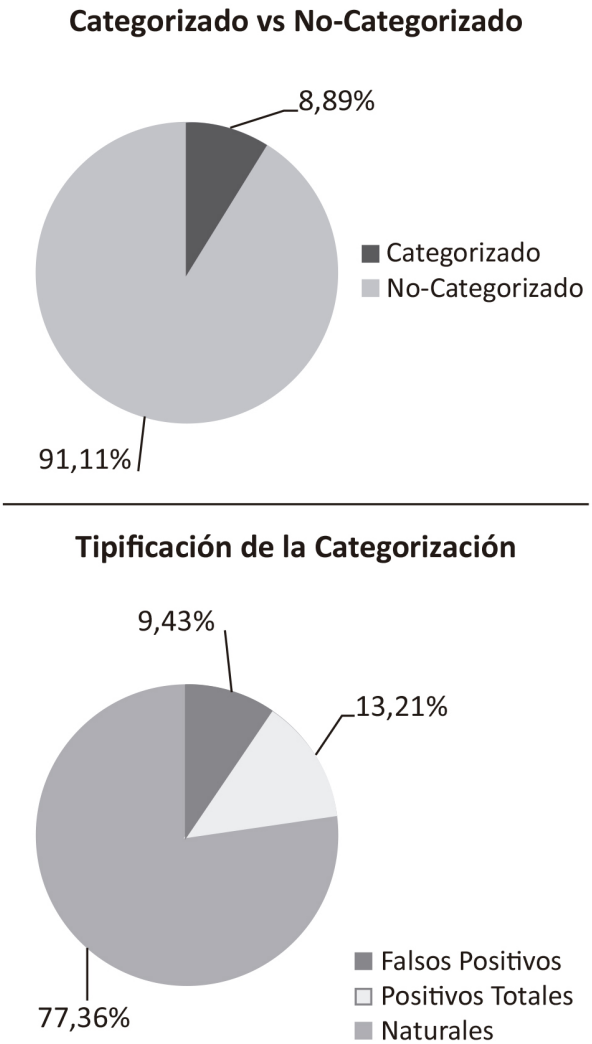

Figura 9 - Resultado comparando las tablas categorizadas contra las no categorizadas (arriba). Resultado de las tablas categorizadas, separadas por tipo (abajo).OpenERP.

Sin embargo este ERP tiene un detalle que es importante mencionar: muchas palabras en las etiquetas de columnas y tablas fueron escritas en

francés, en lugar de inglés. Para analizar la incidencia del idioma francés en la BD se separaron las palabras manualmente (corrigiendo aquellas separaciones incorrectas) y se obtuvieron los siguientes datos:

- Total de palabras en nombres de tabla: 569.
- Palabras en francés en nombres de tabla: 126.
- Total de palabras en nombres de columna: 3235.
- Palabras en francés en nombres de columna: 307.

Calculando hay un 22.14% de las palabras en los nombres de tablas escritas en francés y un 9.49% en los nombres de columnas. Analizando el ERP completo se obtiene que el 11.38% de las palabras empleadas en la BD fueron escritas en francés en lugar de inglés.

Para trabajar con estas palabras el procedimiento fue distinguirlas y hacer una lista con los significados en inglés de cada una y posteriormente agregar las palabras francesas al archivo de sinónimos de la palabra en inglés. De este modo no se modificó la base de conocimiento ni las bolsas de palabras pero se le dio a GrACED la capacidad de comprender (limitadamente) el francés.

Una vez completados los archivos de sinónimos Dolibarr fue analizado con GrACED y se obtuvieron los resultados que pueden verse en la Figura 10.

De este análisis surge que el 6.82% de las tablas de Dolibarr contienen información de la categoría Product Definition que es lo que puede observarse como la porción denominada categorizada en la Figura 10. Considerando las categorizaciones realizadas en los tipos definidos previamente, en la Figura 10 abajo se observa que el 78.57% son de tipo Neutral, el 7.15% son Positivos Totales y el 14.28% restante fueron consideradas Falsos Positivos.

Nuevamente, y con el objetivo de evaluar el comportamiento de GrACED en el proceso, se comparan los resultados automatizados contra la clasificación manual realizada por expertos. Así los expertos realizan 16 categorizaciones y GrACED coincidió con 13, lo que representa un 81.25% de certeza. A su vez el agente agrega sólo 1 categorización, la cual fue posteriormente aceptada como correcta por los expertos.

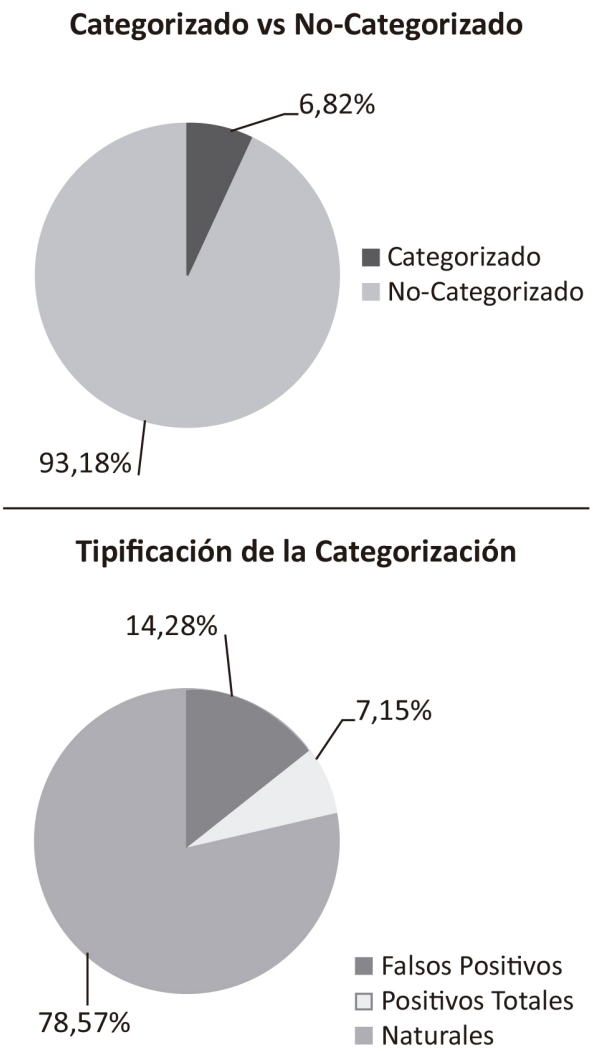

Figura 10 - Resultado comparando las tablas categorizadas contra las no categorizadas (arriba). Resultado de las tablas categorizadas, separadas por tipo (abajo).Dolibarr.

Este caso de estudio se considera exitoso, si bien es importante destacar la diferencia de tamaño de las bases de datos de OpenERP y Dolibarr, dado que este último tiene una BD de un tamaño 60% menor, aproximadamente. Esto deriva en me nor cantidad de tablas con información de Definición del Producto y, por ende, menos clasificaciones.

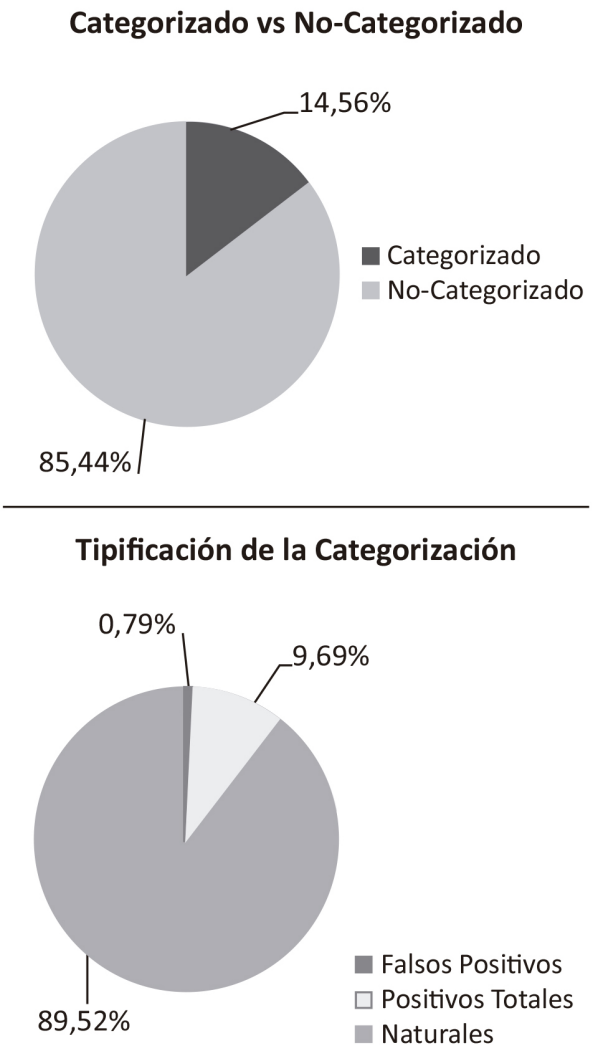

Figura 11 - Resultado comparando las tablas categorizadas contra las no categorizadas (arriba). Resultado de las tablas categorizadas, separadas por tipo (abajo).Adempiere.

ADEMPIERE

Adempiere (Pamungkas, 2009) es otro ERP de código abierto, se desarrolló como un fork1 de Compiere y fue publicado bajo una licencia GNU General Public License (Free Software Foundation Inc., 2007). Este sistema tiene una base de datos de gran tamaño implementada en Oracle 10g XE.

Este ERP tiene una gran base de datos (con 726 tablas y más de 14000 columnas) pero no posee una buena precisión en la separación de palabras ya que no ha empleado una convención en particular. Esto puede observarse a continuación:

- Total de nombres de tablas: 726.
- Nombres de tablas correctos: 316.
- Total de nombres de columnas: 14033.
- Nombres de columnas correctos: 4958.

Con estos datos se obtiene que sólo 43.52% de los nombres de tablas tienen una separación correcta, mientras que en las columnas el porcentaje es aún menor, apenas alcanzando el 35.36%.

Esto sucede debido a que no hay una estandarización en la utilización de un naming conventions dentro de la implementación de esta BD: la única separación que puede considerarse como tal es la utilización del guion bajo debido a que posee varias repeticiones.

Otro problema en esta base de datos es el uso de palabras genéricas como “bname” o “description” sin emplear otros modificadores que agreguen mayor valor semántico, lo que disminuye considerablemente el porcentaje de pertenencia obtenido al intentar clasificar las tablas de ésta. Finalmente otro problema de Adempiere es la redundancia de información: se encuentran repetidas gran cantidad de tablas que guardan la misma o similar información y que sólo agregan datos duplicados, dificultad de mantenimiento y de integración con otros sistemas.

Se analiza Adempiere con GrACED con el objetivo de poder estudiar el comportamiento del agente en ambientes que no son óptimos. Los resultados de este estudio se encuentran en la Figura 11.

De los resultados obtenidos se puede ver que el 14.56% de las tablas de la BD de Adempiere contienen información de Product Definition generando 93 clasificaciones. A su vez tipifica las clasificaciones de la siguiente forma: el 89.52% son

1 Un fork sucede cuando los desarrolladores copian el código fuente de un paquete de software y comienzan un desarrollo independiente sobre éste, creando un software distinto. Común en desarrollos de código abierto.

Neutrales, el 9.68% son Positivos Totales y sólo el 0.79% son Falsos Positivos.

Sin embargo el impacto de la redundancia y la incoherente separación de palabras se ve reflejado en las coincidencias: los expertos sólo clasificaron 30 tablas, de las cuales GrACED coincidió con 19 (un 63.33%). No obstante el agente agregó 73 clasificaciones y sólo 13 fueron consideradas como adecuadas por los expertos; esto significa que el 65% de las categorizaciones fueron agregados inconsistentes del agente.

En este caso de estudio, y ante la presencia de problemas de redundancia, carencia de naming conventions y utilización excesiva de palabras no representativas, GrACED sobre-categoriza las tablas en lugar de clasificar de menos. Esto se debe a lo siguiente: un nombre de columna como ismanufacturingresource debería contar como 3 palabras pero, en realidad, el agente la distingue como una sola palabra porque no la puede separar. Esto hace que la cantidad de palabras totales sea menor que las reales y se pasen los filtros de cantidad de palabras encontradas.

Se puede concluir que es un caso exitoso porque se sostienen las premisas que se plantearon al inicio: al trabajar en lenguaje natural existe una fuerte dependencia entre la separación de las palabras y los resultados de la clasificación así como también entre la semántica de las palabras empleadas y las bolsas generadas.

CONCLUSIONES

El presente trabajo propone la estructura básica para un agente inteligente basado en conocimiento denominado GrACED el cual trabaja con lenguaje natural (idioma inglés) y que utiliza una base de conocimiento bolsas de palabras generada a partir de la estructura de datos, modelos y definiciones de categorías de información de manufactura propuestos en el estándar ANSI/ISA-95.

El objetivo de GrACED es enlazarse con un sistema ERP (reconocido como su ambiente) y analizar el contenido de su base de datos para estudiar no sólo cómo se estructuran los datos sino también para encontrar la información necesaria para la integración entre sistemas. Esto resulta especialmente útil al momento de la integración de sistemas de los miembros de una cadena de suministro o al intentar lograr la colaboración entre el sistema empresarial y un sistema tipo APS (Advanced Planning and Scheduling).

Su funcionalidad fue evaluada a través de tres casos de estudio empleando sistemas ERP de código abierto: OpenERP, Dolibarr y Adempiere, logrando comportamientos favorables con una precisión total mayor al 80% en los casos exitosos y un comportamiento esperado en el caso negativo.

Como prototipo del proyecto la implementación actual de GrACED ha logrado buenos resultados por lo que surgen varios trabajos futuros, entre ellos, lograr la utilización completa del grafo de clasificación y evaluar un caso más de estudio: Libertya, un ERP de código abierto de origen argentino y base de datos completamente en español.

Otro punto importante es lograr la propagación de pertenencia a las distintas categorías. Observando el grafo de la Figura 3 la propuesta es que, utilizando las pertenencias obtenidas en la clasificación básica desarrollada en este trabajo, se pueda propagar el porcentaje hacia arriba en el grafo con el objeto de encontrar el impacto que cada tabla tiene en el total de la información de manufactura contenida en la base de datos. A su vez, dado que una tabla puede pertenecer a más de una categoría, esto serviría para dar mayor información sobre a qué categoría de nivel uno pertenece con mayor intensidad.

Al lograr una pertenencia total también puede estudiarse la adecuación de la base de datos al ANSI/ISA-95, lo cual es el objetivo último de este proyecto ya que permitiría aplicar a GrACED para

estudiar los ERPs que, por ejemplo, desearían aplicarse a una empresa u obtener indicaciones sobre dónde se encuentra la información necesaria para un intercambio de datos estandarizado.

REFERENCIAS

EU-Commission. MANUFACTURE - A vision for 2020. Assuring the future of manufacturing in Europe. Office for Official Publications of the European Communities, (2004).

Harjunkoski, Nyström y Horch, «Integration of scheduling and control - Theory or practice?» Computers and Chemical Engineering 33: 1909-1918 (2009).

Muñoz, Edrisi, Capón García, Espuña y Puigjaner, «Ontological framework for entreprise-wide integrated decision-making at operational level.» Computers and Chemical Engineering 42: 217-234, (2012).

ISA. ANSI/ISA-95.00.01-2000. Enterprise-Control System Integration. Part 1: Models and terminology. ISA, (2000).

Prades, Romero, Estruch, García Domínguez y Serrano, «Defining a Methodology to Design and Implement Business Process Models in BPMN according to the Standard ANSI/ISA-95 in a Manufacturing Enterprise.» The Manufacturing Engineering Society International Conference, MESIC 2013 63: 115-122, (2013).

Kardos, Csaba, Popovics, Kádár y Monostori, «Methodology and data-structure for a uniform system's specification in simulation projects.» Editado por Elsevier. Forty Six CIRP Conference on Manufacturing Systems 2013. 455-460, (2013).

He, Lobov y Matínez Lastra, «ISA-95 Tool for Enterprise Modeling.» ICONS 2012: The Seventh International Conference on Systems. 83-87 (2012).

Vidoni y Vecchietti, «E2OL: Sistema de Planeamiento y Scheduling Personalizable e Integrable con ERPs.» 1º Congreso Nacional de Ingeniería Informática y Sistemas de Información. Córdoba, (2013).

Quiñonez Gámez y Camacho Velázquez, «Validation of production data by using an AI-based classification methodology; a case in the Gulf of Mexico.» Journal of Natural Gas Science and Engineering 3: 729-734, (2011).

Fu, Ke y Mostafa, «Automated Text Classification Using a Multi-Agent Framework.» Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries. Denver, Colorado. 157-158, (2005).

Wallach, «Topic Modeling: Beyond Bag-of-Words.» Proceedings of the 23rd International Conference on Machine Learning. Pittsburg, PA, 977-984, (2006).

Norvig y Russel, Artificial Intelligence: A Modern Approach. Prentice Hall, (2010).

ISA. ANSI/ISA-95.00.03-2005. Enterprise-Control System Integration. Part 3: nActivity models of manufacturing operations management. ISA, (2005).

W3C Recommendation. Extensible Markup Language (XML) 1.1 (Second Edition). 2006. http://www.w3.org/TR/xml11/#sec-xml11 (último acceso: 01 de 04 de 2014).

Butler, Wermelinger, Yijun y Sharp, «Relating Identifier Naming Flaws and Code Quality: An Empirical Study.» 16th Working Conference on Reverse Engineering. Lille. 31 – 35, (2009).

Roa, Gutiérrez, Pividori y Stegmayer, «How to develop intelligent agents in an easy way with FAIA.» Cap. 4 de Quality and Communicability for Interactive Hypermedia Systems: Concepts and Practices for Design, 120-140. IGI global, ed. Francisco V. Cipolla Ficarra, (2010).

Kohavi, «A study of cross-validation and bootstrap for accuracy estimation and model selection.» Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence. Santa Mateo, CA. 1137–1143, (1995).

Pretorious, Compiere 3. Birmingham: Packt Publishing Ltd., (2010).

OpenERP S.A. OpenERP. Vers. 7.0. 2012. https://www.openerp.com/ (último acceso: 20 de April de 2014).

Panorama Consulting Solutions. «ERPNext.» 19

de Noviembre de 2010. http://panorama-consulting.com/erp-vendors/erpnext/ (último acceso: 2014).

NightLabs Consulting GmbH. JFire . 2011. http://www.jfire.net/ (último acceso: 2014).

Destailleur, Dollibar ERP/CRM. Vers. 3.5.3. 2014. http://www.dolibarr.org/ (último acceso: 2014).

Cristina, Mauprivez, Nerón Cap, Castro y Bonafine, Libertya ERP. 2011. http://www.libertya.org/producto/preguntas-frecuentes (último acceso: 2014).

Pamungkas, Bayu Cahya, «ADempiere 3.4 ERP Solutions.» Birmingham, UK: Packt Publishing, (2009).

GNU Affero, Affero General Public Licence. Vers. 3. 2007. http://www.gnu.org/licenses/agpl-3.0.html (último acceso: 20 de April de 2014).

Free Software Foundation Inc., GNU General Public Licence. Vers. 3. 29 de June de 2007. https://gnu.org/licenses/gpl.html (último acceso: April de 2014).

Brandl, «Business to manufacturing (B2M) collaboration between business and manufacturing using ISA-95.» Revue de l' electricite et de l' electronique, nº 8: 46-52, (2002).

Fecha de Presentación: 15/05/15
Fecha de Aprobación: 08/06/15