Computers in Industry 73 (2015) 39–50

Contents lists available at ScienceDirect

Computers in Industry

journal homepage: www.elsevier.com/locate/compind

An intelligent agent for ERP's data structure analysis based on ANSI/ISA-95 standard

Melina C. Vidoni, Aldo R. Vecchietti *

Ingar CONICET-UTN, Avellaneda 3657 P1, 3000, Santa Fe, Argentina

ARTICLE INFO

Article history:
Received 15 December 2014
Received in revised form 18 May 2015
Accepted 24 July 2015
Available online

Keywords:
ERP
Intelligent agent
Standard
ANSI/ISA-95
Manufacturing information

ABSTRACT

This paper presents an intelligent agent to analyze the ERP's (Enterprise Resource Planning) system data structure and its compliance on the ANSI/ISA-95 standard. The knowledge base of the agent is generated using the manufacturing categories information provided by mentioned standard. The approach proposes an infrastructure of a knowledge-based agent that interacts with the database of an ERP system, in order to classify the information of ERP's database tables according to the standard. Several study cases are evaluated and the results obtained are shown in different graphs. This is a first step to improve the interoperability between an Advanced Planning and Scheduling (APS) system that needs to be integrated with ERP's especially in manufacturing and production companies.

1. Introduction

Enterprise Resource Planning (ERP) has emerged as a *de facto* standard in the market of the generic information systems for many companies [1]. ERP provides an integrated platform to manage the enterprise business in all sectors where information automation is possible. Since ERPs are general systems that can be applied to companies having different characteristics: retailer, production, manufacturing, services, etc.; they need a customization step according to the company business in order to get its benefits [2]. Manufacturing and production companies require also an Advanced Planning Systems (APS) in order to manage production planning and scheduling to optimize material and human resources, improve the company economy and offer a good customer service [3]. In general, ERP does not provide a good functionality for that purpose, there is a gap between what ERPs offer and what the companies' needs [4]. APS are in general *ad hoc* applications that complement several functionalities of the existing ERP systems. It is a common practice that implementations of an APS in the ERP or company system are made in an improvised way, involving several stakeholders, consultants and internal people, without a methodology and tools to guide this process and therefore, there is a high interest in a better understanding of the success and failure factors on the implementation of this type of software [5].

In the last years, another issue that becomes a key factor for enterprise success is collaboration between organizations which enables the companies to enforce their partnership to strengthen their business in the market. This is done by generating a standard and interoperable communication between their systems and applications [6]. As a consequence, organizations face a lack of interoperability of their current systems [7], mainly due to incompatibility problems in the information representation and in the adopted software application methods [8]. This situation affects the development and integration of custom made systems to the current structure, which is deeper in cases where the new modules are related to production operations and process related data [9].

In response to this fact, the European Commission recommended the improvement of the integration process through their standardization and further automation [10]. In order to overcome the integration, it is mandatory to define the information structure and tools, with the aim of improving data availability and communication, more specifically in the company's supply chain [11]. Many standards are recommended along these lines. ANSI/ISA-95 (also known as S95) is an international standard to develop automated interfaces between organizations and their control systems, proposing a set of models and definitions to describe the tasks and manufacturing and production information that must be exchanged between information systems [12]. In the last years, this standard has been widely accepted given its complete functional model [13].

* Corresponding author at: Instituto de Desarrollo y Diseño INGAR CONICET-UTN, Avellaneda 3657, 3000 Santa Fe, Santa Fe, Argentina.
E-mail addresses: melinavidoni@santafe-conicet.gov.ar (M.C. Vidoni), aldovec@santafe-conicet.gov.ar (A.R. Vecchietti).

http://dx.doi.org/10.1016/j.compind.2015.07.011
0166-3615/

The incremental application of these standards on the industry has influenced the academic environment, and there are many works focused in the exchange of standardized information, using the models of ANSI/ISA-95. In 2006, there is a proposal for a standard-based extendable platform, to support an interoperable environment through the adoption of MDA (Model Driven Architecture) and SOA (Service Oriented Architecture), including several ISO standards [7]. Later in 2009 [14], Harjunkoski et al. proposed a framework for information exchange using BPMN (Business Process Model and Notation) diagrams based on the models of ANSI/ISA-95. He et al. [15] developed a tool based on ANSI/ISA-95 and IEC 62264 standards for enterprise modeling. Nagorny et al. [16] generated an approach for developing and implementing a service and multi-agent oriented manufacturing automation architecture, focused on features of the IEC 62264 L2 standard; the goal behind their proposal is to facilitate the managing and control of networked smart automation components in a distributed manufacturing environment.

From the previous paragraphs, it can be seen that while there are many works and frameworks focusing on developing new systems, according to several standards, to our best knowledge, the current research done on studying and analyzing the existing data structure and system's functionality to evaluate the suitability to a certain standard or a particular classification is not abundant.

There have also been advances in the application of artificial intelligence techniques in the area of manufacturing and system interoperability. It can be quote Clover [17], an agent-based cooperative intelligent design environment with a focus on the issue of systems interoperability that uses several ISO standards. Also, a review in 2006 [18] discussed some key issues in the implementation of agent-based manufacturing systems such as agent encapsulation, tools and standards. The authors remark the importance of the integration is between existing ERP and MRP (Material Resource Planning) systems.

This work presents an intelligent agent based for classifying and studying an ERP's data structure, using natural language. This agent is a knowledge-based type [19], and it is constructed upon the categories for manufacturing information proposed on ANSI/ISA-95 [12]. In order to study the data structure of an ERP, the agent processes natural languages through the use of a bag-of-words [20] approach, with several modifications to consider importance of words, and use of synonyms. Finally, the agent was implemented using FAIA [21], a Java-based framework to develop intelligent agents.

The aim of this work is to provide a tool to analyze if the ERP's data structure is in compliance ANSI/ISA-95 specifications. This intends to be a first step to overcome the gap existing in the integration between ERP's and APS's systems and also with other information systems intra/extra company in the supply chain, without forcing organizations to a radical change in their information systems.

This paper is organized as follows. Section 2 introduces the initial study of ANSI/ISA-95 and the development and implementation of the knowledge base of the intelligent agent, addressing several issues pertaining to it. After that, Section 3 presents the development of the agent's main structure, actions and reasoning algorithms; this section also discusses statistical studies done to adjust and improve the agent's behavior. Finally, Section 4 introduces several study cases, executed using open-source ERPs, that allow validation of the agent's behavior and results.

2. GrACED: knowledge-based agent

The agent proposed on this work is named GrACED, an acronym that stands for 'Grammar Agent for Classifying ERP Databases', and is developed following mainstream definitions for *intelligent agent*. Russell and Norvig [19] defined intelligent agent as an autonomous entity inserted on an environment that observes what happens on it through perceptions (made with sensors), and responds to them by performing actions with actuators. There are many types of agents, some of them can learn on their own (learning type) while others have a goal to reach (goal-based type).

Knowledge-based (KB) agents are special types of the previous definition. They possess knowledge representation (usually on a so-called *knowledge base*) and a reasoning process that executes and combines with perceptions, before selecting more actions [19]. These types of intelligent agents are very useful to process natural language, because they are able to understand the semantic behind the words.

2.1. ANSI/ISA-95 study

ERPs information is stored in its databases (DB), which, currently, are commonly relational-type [22]. Due to this reason, one way to study the information organization of an ERP is to analyze and classify the data structure of its database tables. For this purpose the ANSI/ISA-95 standard is used, which proposes models and consistent definitions of manufacturing and production information [12]. More precisely, in the Part III [23] the standard describes four categories to define the products and production information, which are part of the knowledge base of the intelligent agent proposed in this work.

The four mentioned categories can be seen in Fig. 1.

In this paper, only *Product Definition*, *Production Capability* and *Production Schedule* are employed, because these are the most populated categories containing relevant information to be used in GrACED. Also, since the goal is to classify ERPs DB, the data of *Production Response Information* category is more likely to be stored in some attributes of the tables (columns) and not in full tables, which would exponentially increase the complexity of the categorization; this is why this category is not selected to be part of the knowledge base of the agent.

Part I of the standard [12] also provides definitions and concepts that are used to generate the structure and outline the categories that are used by the agent to classify the information. The standard proposes graphics that are called 'overlay charts', because they show subcategories with overlapped information between each other. Fig. 2 shows the overlay chart for the main category *Production Definition*, which presents the overlap among *Product Production Rules*, *Bill of Materials* and *Bill of Resources*.

Fig. 3 shows the categories and subcategories corresponding to ANSI/ISA-95 standard used to define the knowledge base of the agent. The oval nodes represent the categories while the rounded rectangles are subcategories used to classify the ERP's information.

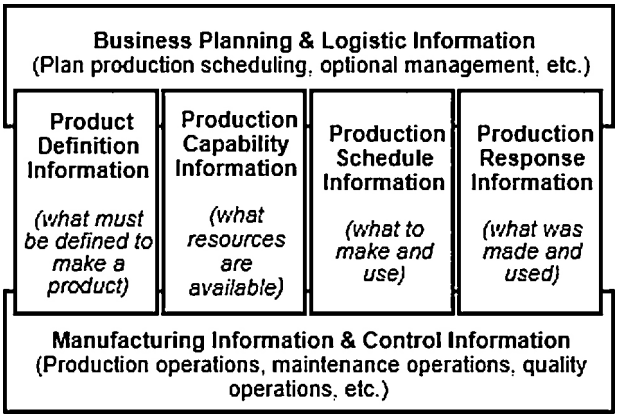

Fig. 1. Information categories proposed on ANSI/ISA-95.

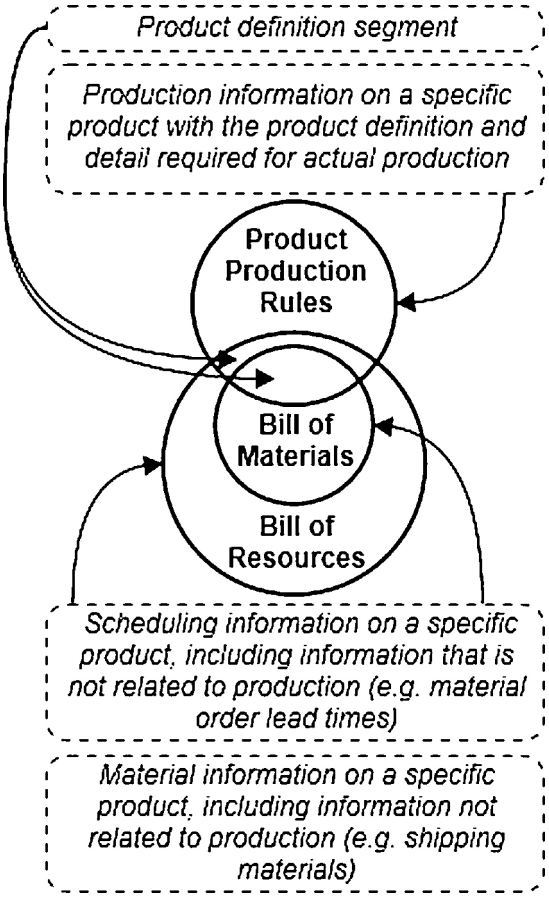

Fig. 2. Overlay chart for the *Production Definition* category, on the Part I of ANSI/ISA-95 standard.

From Fig. 3, the root node denotes the whole set of Manufacturing Information (MI), whereas the level 1 nodes are the main categories of Fig. 1 (without using *Production Response*, as previously stated). On the other hand, the lower level nodes (level 2, and lower) are obtained from the overlay charts and the category descriptions presented in Part I of the standard.

For example, the subcategories presented in Fig. 2 are child nodes of *Product Definition* in the graph of Fig. 3. From this figure it can be seen that because Bill of Material (BoM) is overlaid with Bill of Resources (BoR) it is represented in the graph as a child node.

2.2. Bags of words

Regardless of the implemented Database Management System (DBMS) the tables and columns have names that give them a semantic meaning of the stored content. Therefore, the selected approach for classifying the DB content is based on *bags-of-words* (BoW), which is a simplified representation used for processing natural language, where each class or document is depicted on a multi-set (or bag) of words, considering nor the grammar (forming sentences) nor the order of words [20]. It is worth mentioning that the ANSI/ISA-95 only describes the categories, noticing what type of information they include, but without supplying any words to create a BoW.

However, not all the words have the same relevance, which can also vary from category to category; because of this fact, the BoW approach is combined with *weights*, giving a certain value to the words inside the bags. Since all the bags are equally important, each of them has a total weight of 100, which is internally splitted between words, giving a bigger value to the most vital ones on each category.

Frequently, during the development of a database, the words used to define its table names are often not the same as those used in the column names, even when they belong to the same category. As a consequence, a decision is made to associate two BoW per category: one for the words that may appear in the table names and the other for the columns.

For example, considering only the BoW for table names, the word "`product`" has a weight of 15 in the *Bill of Materials* bag, a weight of 20 in *Bill of Resources* bag, and 10 in *Production Rules* bag; another example can be the word "`bom`" that has a weight of 25 in *Bill of Materials*, but does not have weight on the other two because it is not part of the other bags.

An important detail is to consider the use of synonyms or abbreviations employed at the moment of naming tables and columns. It is not convenient to add each possible combination for each word directly to the main BoW, because not only adds redundancy but exponentially increases the processing time, reduces the weight of the words inside the bags, and can also have a negative effect in the final belonging percentage of a table. As an example, both "`product`" and "`production`" can be shortened as "`prod`". To solve this situation, exclusive files – named *Synonyms Files* – containing the synonyms and abbreviations are linked to each word on each BoW. These files contain a match between a *main word* and all of its synonyms or abbreviations, allowing the flexibility and variety of natural languages but avoiding redundancy and weight-reduction on the main BoWs.

In this point, it is noteworthy that since the chosen natural language is English; because of this, the use of a different language is considered as a "synonym" of English, and it is added to the Synonym Files, instead to the main bows. This situation is shown in one of the study cases.

2.3. Knowledge base implementation

As mentioned previously, two BoW are linked to each node of Fig. 3 graph. Such graph works as the index on the knowledge-base of the agent (see Fig. 4), storing all the references to the categories and the bags of words, but keeping the hierarchy between the nodes and their levels. Also, each word on each BoW may point out to a *Synonyms File*, where all the alternative words with the same meaning, are stored.

In order to test the first implementation, only a sub-set of the nodes in graph of Fig. 3 is employed; this decision is done to simplify testing the approach and also to be more manageable.

The selected node to translate into the BoW is *Product Definition* (and its subtree); also, only the subcategories of *Bill of Materials*, *Bill of Resources* and *Production Rules* are employed for the classification action. This condensed graph can be seen in Fig. 5.

With the purpose to generate the knowledge base, four open-sources ERPs are selected, which are: Compiere [24], OpenERP [25], ERPNext [26] and JFire [27]. The implementation is done using eXtensive Markup Language (XML) [28].

Table 1 shows the final numbers that quantify the size of the KB. These numbers include only raw words inside the bags-of-words, and do not count alternative words on the Synonym Files.

The process to generate the BoW and Synonyms files is done only once, manually, and following these steps:

1. For each selected ERP:
   a. List tables and their columns.
   b. A selected group of experts categorize the tables, considering the scope definitions for each category existing on the ANSI/ISA-95 standard. The tables which do not contain information related to the *Product Definition* class are not categorized.

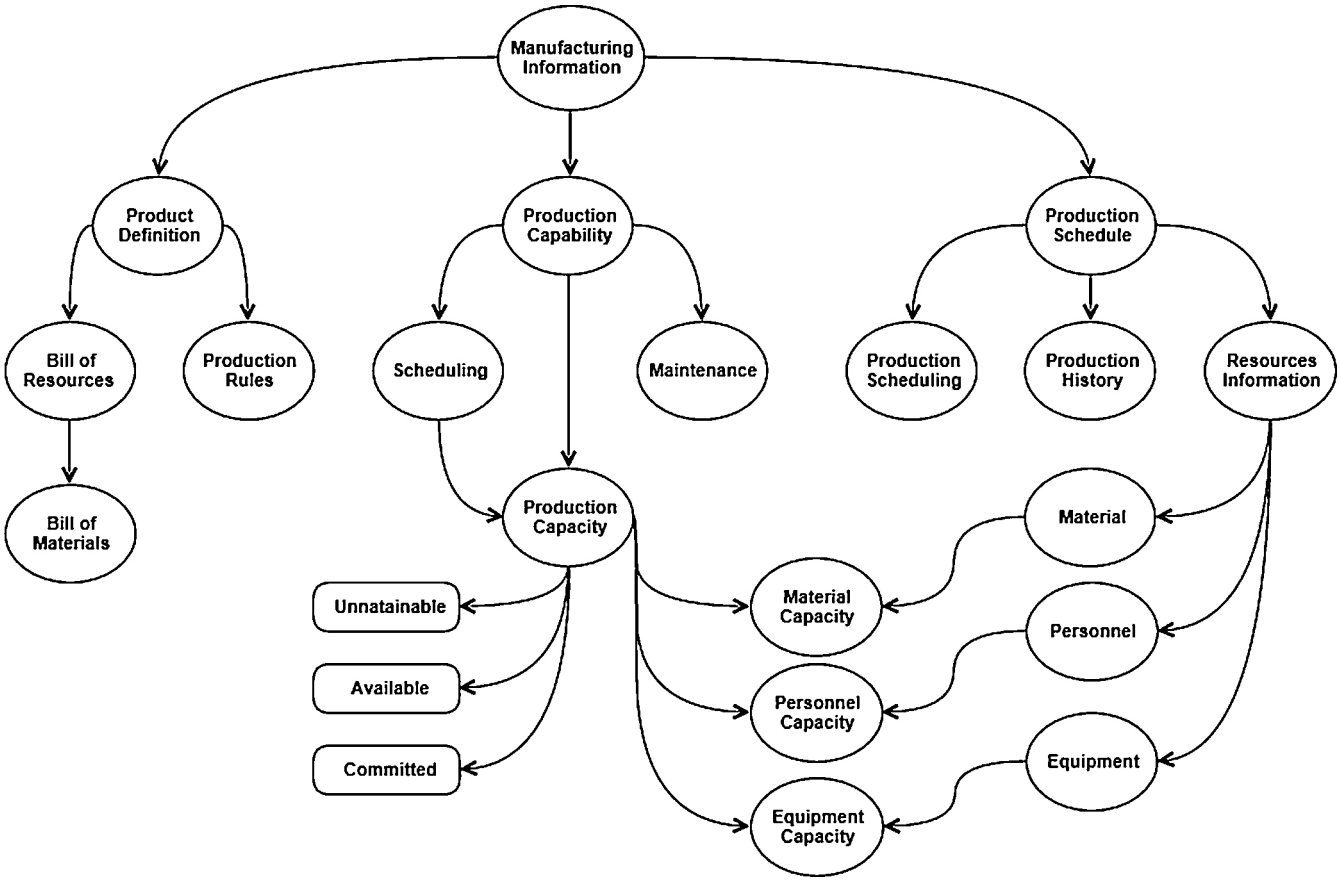

Fig. 3. Categories' graph derived from ANSI/ISA-95 Part I.

c. Tables names and columns names are manually splitted into words. For example, the table name `stock_inventory_move` is splitted into three words: `stock`, `inventory` and `move`.
2. Keeping the source of the words, meaning if they came from table names or from columns names, they are grouped by category.
3. For each group of words:
   a. Each word is associated to its synonyms and abbreviations.
   b. The number of times each word and its alternatives appear is counted to obtain the relevance (or importance level) to a category.
   c. The number of appearances of each word is used to obtain the final proportion in the total weight of the bag.

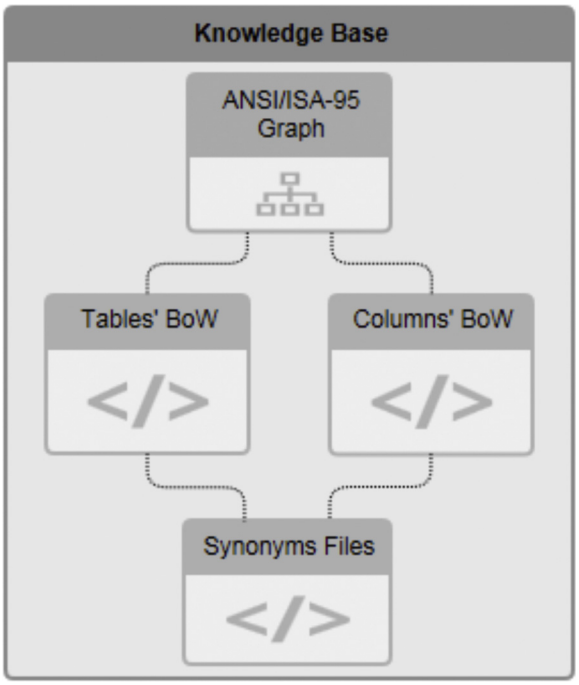

Fig. 4. Proposed KB structure, based on the ANSI/ISA-95 categories.

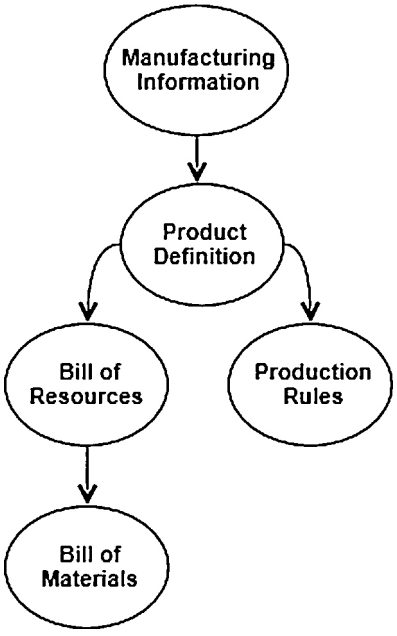

Fig. 5. Initial implementation of the original ANSI/ISA-95 based category graph.

Table 1
Information regarding the size of the KB implementation.

TablesBoW quantity (TBoW)	3
ColumnsBoW quantity (CBoW)	3
Synonyms Files quantity	189
Total Words on TBoW	70
Total Word son CBoW	423
Column names vs Table names proportion	6043
Total Words on BoW	493

d. The alternatives are written on Synonyms Files for each category, and only one appearance of the "main word" is kept.

3. GrACED: algorithms and structure

Following the main components of the definition stated in the beginning of Section 2, Fig. 6 shows the basic structure of GrACED, focusing on the algorithms and component structure of the agent.

GrACED is inserted into an environment corresponding to target ERP to be analyzed. This environment has a state, composed of the list of tables and the information needed to have a connection with it.

Furthermore, the agent has two perceptions that are interrelated: one for the tables name to analyze, and the other for the columns names. These perceptions are stored in the agent state while it executes the following actions:

1. *Classify*: this action reads the knowledge-base and the available nodes for classification, matching the words with the synonyms and analyzing if it belongs to a category or not.
2. *Hyphenation*: this action estimates separation correctness of each word using a fixed number. This is done with the purpose of having an approximation of how good is the naming rule of the ERP database.

The other components of the agent states are: a temporal list of the classification obtained when analyzing the current table; and a link to the knowledge base.

Due to the implementation of an intelligent agent is a complex task, its development is done using FAIA [21]: a Java based framework that offers a structure of abstract classes to implement several types of agents (reactive, goal-based, knowledge-based, etc.). FAIA provides a structure to program the basic functionality of the agent (the entity, the environment, the states, perceptions, and actions). The full development is made in object-oriented Java8 with graphical interfaces done with JavaFX.

3.1. Classification action

This action executes a reasoning algorithm to match each table with one or more ANSI/ISA-95 categories. The agent uses the "enabled" graph nodes trying to match the information of each database table. It is noteworthy that the table cannot be categorized in case it does not contain manufacturing information or belongs to more than one group. The reasoning algorithm is presented using the flowchart of Fig. 7, where diamonds are either decisions having conditions (or '*filters*') to establish if a categorization is worthy or not, or loops.

The first conditional diamond compares the quantity of words of a table against the words in the BoW; in order to move to the next filter, at least half of the words have to be found in a BoW. For example, the table named `mrp_production_product_line` is splitted in four words (`mrp`, `production`, `product` and `line`), and at least two of them must be found in a BoW so that the agent can keep the category and goes to the next filter.

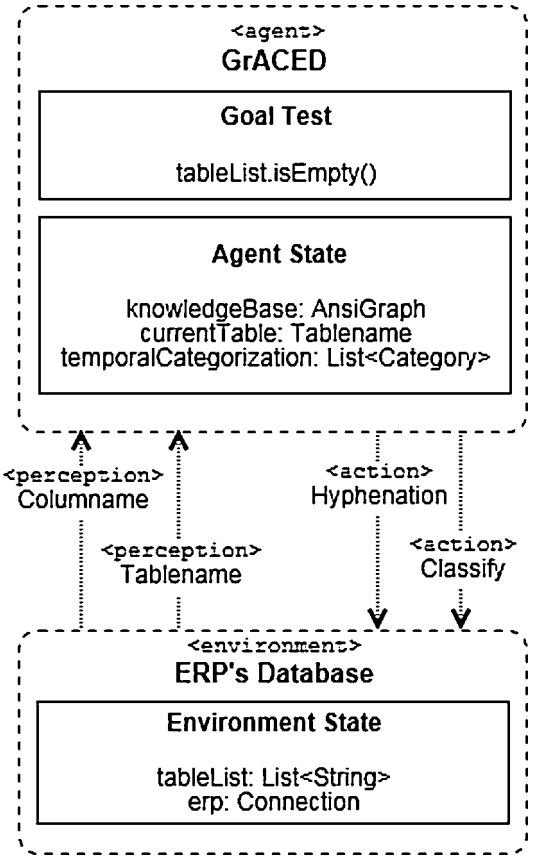

Fig. 6. Basic structure of the intelligent agent.

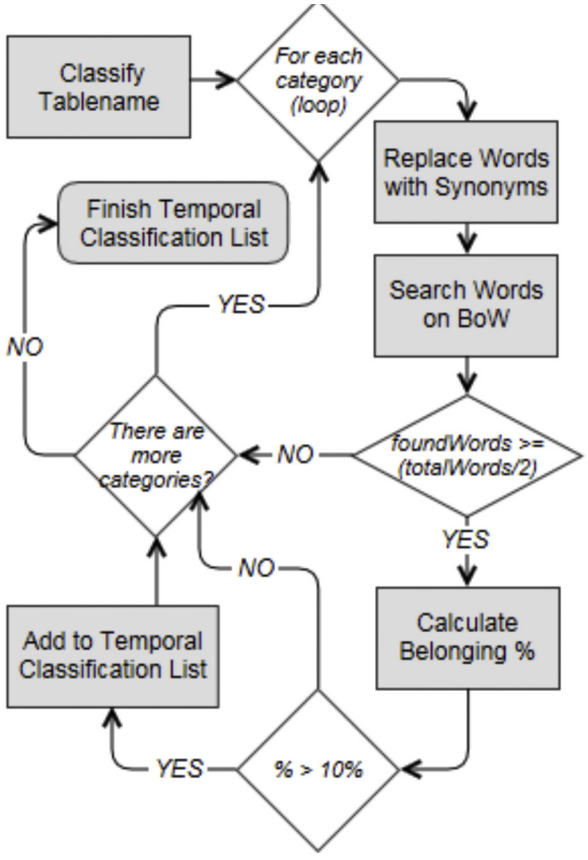

Fig. 7. Part I of the reasoning algorithm's pseudocode: table pre-classification.

The second one implies the calculation of the belonging percentage as follows: given that each bag has a weight of 100, the sum of the word weights returns the percentage; then it is compared to a minimum *barrier number* of 10%. If the percentage is less than 10%, the temporal classification is discarded and the agent continues with the next category, on the contrary, the algorithms continues with the next step (see Fig. 7).

On average, a table name does not usually contain more than four words, because extremely long names also reduce code readability [29], and a column name commonly has 3 or 4 words on its identifier; also, the produced bags-of-words for table names have between 25 and 30 records each, because they have to represent a whole category that has a certain diversity of words. If we consider an example of an ideal case were a table name composed of 4 words matches them all to a BoW of 25 words, it will only have a 16% of belonging, and 13% if the BoW has 30 words, because the difference between the number of words in the table name and in the bag-of-words is quite big; this does not make the table less relevant, but considerably reduces the range in study to select the *barrier number*. In conclusion, a table name with a belonging of 10% is already important to the category where it was categorized.

The second step on the classification is to evaluate the column names, but only using the categories that surpassed the pre-classification on the table name. The steps are similar to those in Fig. 7, but this time there is only one filter where the belonging percentage must be over a second *barrier number*, which is selected via a statistical analysis, using the ERPs previously selected: OpenERP [25], Adempiere [30] and Dolibarr [31].

In order to get a best-choice barrier number, a study is conducted to evaluate possible filter values among a given range, which extends from 2% to 12%. The first step is to obtain a manual classification of the ERP's databases, from the group of experts. Then, the agent analyzes each database with each configuration, totalizing five runs per ERP. The results are compared with the manual classification performed by experts. The obtained matching percentage can be seen in Table 2.

The conclusions are that sometimes the agent '*over categorizes*' tables compared to the manual classification: when the barrier number is too low, GrACED lowers its 'expectations' of how 'fitted' a table must be to a particular category, thus adding many categorizations that should not be added; however, when the barrier number is too high, GrACED discards categorizations that are adequate, but that are usually defined with 'generic' words (words that have an average weight inside a BoW). This behavior is normal but must be taken into account in the result analysis. In Table 2, the barrier-number breakdown starts with 2%; which is rejected because even if the matching percentage is high, GrACED did too many over-categorizations (from 17 to 120), adding error to the results. Also, values 7%, 10% and 12% are also rejected because the agent had a low matching quality and discarded some categorizations that can be correct.

Finally, 5% is accepted as the best-choice barrier number, because it has a good mixture of matching quality and a low number of over-categorizations. Also, considering that each BoW for column names has around 150 records, the weights are smaller than BoW for tables, and thus the amount of words needed to reach at least a 5% is large.

Table 2
Matching results obtained by evaluating different barrier-number options for the filter in the columns' name reasoning algorithm.

	Matching				
	2%	5%	7%	10%	12%
OpenERP	94.34%	90.56%	58.49%	47.17%	35.84%
Dolibarr	93.75%	93.75%	75.00%	68.75%	68.75%
Adempiere	82.45%	82.45%	64.91%	57.89%	52.63%
Average	90.18%	88.92%	66.13%	57.94%	52.41%

3.1.1. Categorization types

Because the agent has two BoWs per category it keeps separately the belonging percentages to allow a further analysis. The relation between these two values is typified into three types with the purpose to address the magnitude of the words found:

- *True*: covers categorizations where both percentages (table and columns) are above 40%, because the names have many meaningful words with high weights. Few categorizations fit this type.
- *Tricky*: contains classifications where the belonging percentage of the table name is much higher than the obtained through the column names. This happens when the table name has specific words of high weight, while the columns are named using generic ones, with medium to lower weights. A tricky categorization is not discarded because it may contain relevant information.
- *Neutral*: this final type contains categorizations not covered on the above types, where generally both belongings percentages are of a medium level, only containing medium to generic words. These are usually tables that derive from relations in the Entity-Relation diagram of the database and store complimentary data.

3.1.2. Information distribution

While making the categorization, it is important to determine the proportion that the tables contain of each information type, in order to know their uses and the data lacking. For example, a database may contain almost no table to save workflow data, but too many for storing BOM-related information.

After the agent categorizes a table, it propagates the classifications, leaving only one category. This does not mean that a table cannot have more than one category, but that it may be more suitable to one, among all the classifications that surpassed the filters during the main reasoning algorithm. This distribution process has an algorithm that is executed in two parts. The first one is the selection of the base category from all those assigned to the table. This part of the algorithm is graphically represented in Fig. 8 where, once again, diamonds represents choices or loops of the algorithm.

This flow of the code goes as follows: if a table is categorized with only one class, this is selected as the base category. If it has more than one, GrACED evaluates their combined belonging percentages, looking for the biggest one that has also at least a minimum difference with the other categorizations. Finally, if no category has that minimum gap, the algorithm calculates Bayes rule for each one.

As seen in Fig. 3, there are categories that have more than one parent, and in that case, the Bayes rule is calculated for each parent. The probabilities used on this procedure are also obtained in the moment of generating the knowledge-base of the agent.

As expressed, the percentage employed on this part is the combination of both belongings obtained through the analysis of the table name, and of the columns names. The basic formula to determined it is expressed in (1), where *tw* is a weight for the table percentage, and *cw* for the column's.

$$combined\% = wt{*}table\% + cw{*}columns\% \quad (1)$$

Three possible combinations are considered:

- *Equal importance*, meaning that both the table and the columns will have the same weight, because both categorizations are equally relevant. This results on using a 0.5 weight for both, changing the equation to an average.

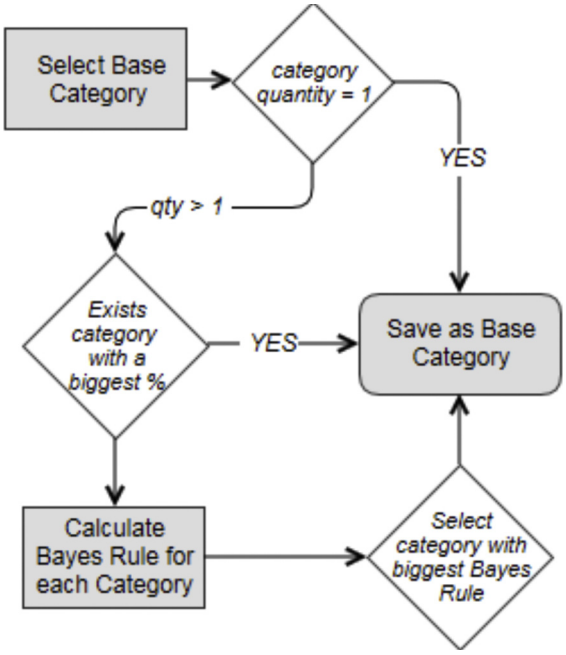

Fig. 8. First part of the distribution algorithm's pseudocode: select base category.

- *Tables are more relevant*, means that the belonging obtained through the table name is more important, hence it will have a bigger weight. The selected proportion is $tw = 0.7$ and $cw = 0.3$.
- *Columns are more relevant*, meaning the opposite of the previous case, the belonging percentage obtained in the column names is more important, and will have a bigger weight: $cw = 0.7$and $tw = 0.3$.

These possibilities are statically evaluated in the second part of the algorithm: *updating up-tree*, whose recursive pseudo-code is graphically presented in Fig. 9.

In this step, the agent continues to recursively update the values, until it reaches the root node. The simplest case is when the current category only has one parent. A more complex situation is when there is multiple-hierarchy, and the agent needs to select which parent to update. For this case it calculates the Bayes rule for each parent category and selects the one that achieve a bigger value, without considering any minimum gap.

Three possible combinations are evaluated on a statistical study altogether with the *minimum difference gap* that the algorithm uses on the first part, in order to find the combination with best value.

Also, three promising values for the gap are considered: no gap, a 2% gap, and a 5% gap. Each one is mixed with the three combination possibilities, generating nine scenarios, which are also studied for each case study, making a total of 27 runs. The results are compared to the default category proportions and the selected base category is contrasted to the manual categorization.

Those studies prove that having no-gap impacts on the propagation up-tree, selecting categories that have a lower bound with its parents (smaller value when calculating Bayes rule). In the other hand, the gap of 2% or 5% impacts only when studied using the formula. In both cases, using a bigger weight on the columns (case: $cw = 0.7$) leads to a wrong selection of base categories while comparing it to the experts'. The "equal importance" and "tables more relevant" case has the better approach to select the base category, and also the best distribution when using a gap of 2%.

This distribution is compared considering the entire implemented categories (nodes in the sub-graph of Fig. 5). For this purpose, Fig. 10 shows the two selected combinations and their

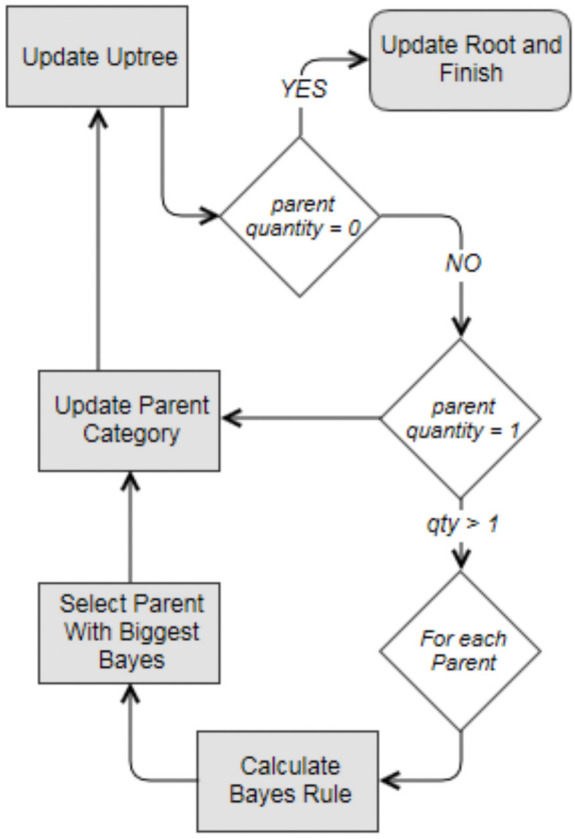

Fig. 9. Second part of the distribution algorithm's pseudocode: update up-tree.

near-default proportions for each ERP: it compares the ratio (*y*-axis, on percentages) of manufacturing information found on each ERP, plus those obtained from a manual evaluation.

It is important to mention that due to the magnitude of an ERP and the amount of data it stores, it is expected to find low ratios; this is not an error, but a consequence that the ERPs intends to integrate the whole organization's processes. It is also worth mentioning that Dolibarr stores almost no information regarding workflows, and that explains its low proportion in the process information.

3.2. Hyphenation action

Words semantic and separation are an important part of an intelligent agent comprehension of natural languages [32]. Whether it is recognizing words form a handwritten text, or from a digital document, the hyphenation is not a trivial problem [33].

The words separation has also a great impact on the results of GrACED and because of that it has been decided to provide a simple method that could give an estimated correctness of word separation. Developers use *naming rules* to name variables, functions and classes while writing in programming languages; the reason to use them are to reduce the effort needed to read and understand source code, and to enhance the appearance of the code [34]. There are several types of these rules and each programming language prefers a different one, but there is no rule that forces developers to use those directives. As might be expected, they are not always used, and thus GrACED may face poor word separation.

In order to make the analysis, the agent connect to the database and ask for the '*predefined*' separation method, that can be selected between Pascal Casing, Camel Casing, using a special character (like underscore, hyphen and so on) or a combination of the previous, with a removable prefix.

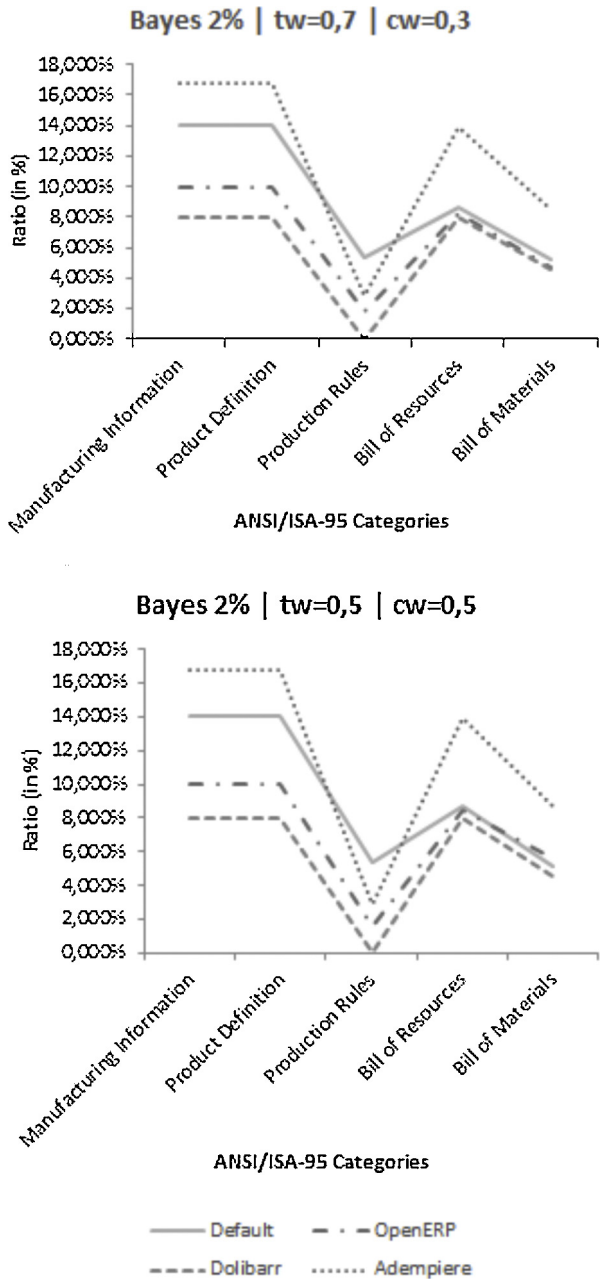

Fig. 10. Mixing gap of 2% with 'tables more relevant' approach (up). Mixing gap of 2% with 'equal importance' approach (down).

That separation is used during the Classify action to split the words. But the Hyphenation action goes beyond that, and compares the splitted words with an *estimated common length* number to estimate if a word is or not correctly hyphenated.

Sigurd et al. [35] studied the frequency of appearance of English words given their length in letters, and concluded that 3-letter words have the highest frequency, while from 7-letters and beyond, the frequency decays until 12-letter words have a occurrence of apparition that is less than 0.917%. Another group of authors made ranged categories, and determined that words of a length from 3 to 5 letters are the most commonly used in English [36]. Considering these values, a statistical analysis is made using an estimated common length from 7 to 12, and comparing the agent result proportions to those obtained manually. This comparison can be seen in Table 3.

The Hyphenation Action results are estimations giving some idea of how good or bad is a naming rule. A proper study would demand an agent with the inherent capacity of understand letter by letter to calculate where each word ends.

Table 3
Hyphenation evaluation results.

	OpenERP	Dolibarr	Adempiere
Matching			
7	0.9054	0.8551	0.6773
8	0.9603	0.8949	0.7454
9	0.9773	0.9419	0.8354
10	0.9889	0.9532	0.8677
11	0.9954	0.9875	0.9148
12	0.9979	0.9928	0.9399
Manual	0.9516	0.8408	0.4701

After inspecting the results, it can be seen that the agent could detect badly separated words if they are long enough; for example, it detected cases such as `alertprocessorlog` or `accounting-transaction` but could not distinguish short combinations such as `mailmsgor` or `salesrep`. These results lead to select an estimated length of 7 because using a bigger number showed correctness proportions that are far from the true words separation.

3.3. Results files

While GrACED continues working, it stores the results on several XML files, using their schema for each case, that allow further review and the generation of graphical charts in the agent GUI. The result files are:

- *Categorized Tables*: this stores the tables that have been categorized on at least one class, saving the category name, and the belonging percentages for tables, columns and the combination.
- *Uncategorized Tables*: contains the tables that do not belong to a categorization. This second file exists because not all the tables of an ERP's DB contain manufacturing information.
- *Hyphenation Estimation*: this file covers the results of the Hyphenation action. It saves the words considering if they are supposedly good or bad separated, and discriminating their origin (tables or columns names).
- *Distribution Information*: for each category of the KB, this file saves the achieved proportion, the default proportion, and the tables that are ultimately associated with it.
- *Category Types*: the semantics behind the words have a high impact on the agent results, and thus the subsequent categorizations are typified onto three types that denote the quantity of generic vs important words, which is typified by the Categorization Types. This file links each resulting categorization to one type.

Additionally, GrACED offers an analytical GUI with graphic charts offering a user-friendly way to study the generated results. Such graphics are:

- Area chart comparing the 'default' information distribution vs the obtained distribution, listing the final base category for each table on a side tree-explorer.
- Pie chart with the proportion of tables containing manufacturing information vs the one that does not (categorized vs not categorized). This is particularly useful to analyze the distribution of the data.
- Another pie chart to showcase the type proportion, and the categorizations that belongs to each type. Those are shown in a table view at the left side of the chart.

- A bar chart for each table, selected from an available list, showing the categorization and its percentages for tables and columns and the average.
- Another bar chart, comparing the estimated hyphenation obtained, disclosing the number of words that are supposedly good or badly separated, and their precedence.

4. Study cases

A group of human experts was selected to perform a manual evaluation of the study cases, and to provide a comparison to GrACED's results. This group consists of four people: the first two are a senior developer with 7 years of experience and a senior DBA (Database Administrator) with 15 years of experience on several DBMS (Database Management Systems). The other two experts are researchers with wide experience on integration of Operation Research (OR). The process to compare the results between the group of experts (GE) and the intelligent agent consisted on the following steps, which are repeated for each study case:

1. The list of tables and their respective columns are given to the GE, and they perform a 'manual' classification, using only the three selected categories (*Bill of Resources*, *Bill of Materials* and *Production Rules*). Each table is allowed to not be categorized, or to have more one- or many-categorizations.
2. GrACED analyzes the same database, and produces a classification.
3. Results from steps (1) and (2) are compared in a spreadsheet that contains: each table name with their corresponding column names, the GE's results, and GrACED's results. This spreadsheet includes the following comparison: matching categorizations, expert-only and agent-only categorizations.
4. The spreadsheet from step (3) is given to the GE. Then, they evaluate their own results and study those obtained from the agent. If they agree with an agent-only categorization, they add it to their manual categorization, in order to turn it to a match.
5. After the GE concludes the revision, the matching percent for the agent is obtained with the results obtained from step (4).

Taking this in consideration, the following subsections presents the study cases used to evaluate the agent's performance, using open-source ERP systems.

4.1. OpenERP

OpenERP [25] is an open source ERP suite, published with an AGPL2 license (Affero General Public License, version 2) [37] and developed as a web application. It is centered in business logic and in the MRP (Manufacturing Resource Planning) module.

The database is implemented in PostgreSQL, has a size of 450 tables, and has a decent consistency at words separation: a manual evaluation determined a 97,405% of correctness, while using the underscore as the predefined naming convention, with all the letters in lowercase. Thus, a name like `m_production_id` is marked as adequately separated, while `movementdate` is considered incorrect, due to the lack of underscore between the words.

This example is analyzed with GrACED, and some of the results can be seen in Fig. 11, that shows two of the charts generated with the agent.

Fig. 11 (top) has the pie chart with the main results, comparing tables that contain manufacturing information (more precisely, from the *Product Definition* level-1 node) and those that does not. In the ERP – installed only with the basic modules – the 9.95% of the tables corresponds to manufacturing information, while the remaining 90.05% does not.

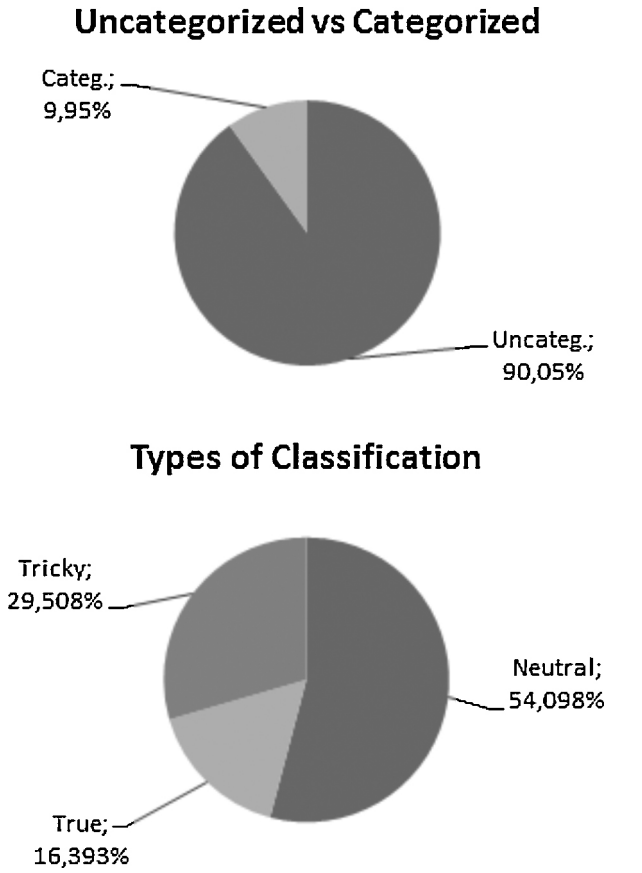

Fig. 11. Pie chart comparing categorized tables vs not categorized (top). Pie chart with categorizations splitted into types (bottom). OpenERP.

The second pie chart (Fig. 11, bottom) shows the categorization types. Here, GrACED counts the total classification because each table may belong to more than one category. Thereby, 54.098% are deemed as Neutral, because most of the words used are generic; a 16.393% is True, using meaningful words, and the residual 29.508% is Tricky.

With the purpose of evaluating the behavior of GrACED, its classification is compared to a manual version done by a group of expert in the area; this comparison results on a matching percentage that compares how many categorizations of the agent are accepted by the group of experts.

After GrACED runs its analysis, it reports 53 categorizations for the OpenERP database; 13 out of those were not previously classified by the experts.

However, all the categorizations from the agent are given back to the experts for feedback, and from the extra 13 categories, the experts agree with 8 more, considering that the agent's results are accurate due to the words used on the database definition of the classified tables.

This makes a total of 48 matching[1] categories and gives a 90,566% of accuracy to GrACED's categorization of OpenERP database.

4.2. Dolibarr

The second study case is done with Dolibarr [31], another open-source ERP published with a GPL (General Public License) version 3.0 [38]. Dolibarr is oriented to medium size enterprises and

[1] Match between the categorizations made by the agent, and the categorizations made by the group of experts, for the same database.

Table 4
Language statistics on Dolibarr database.

Evaluated criteria	Word #
Total words on tables names	569
French words on tables names	126
Total words on columns names	3235
French words on columns names	307

companies, it is a French development, has more than 26 modules, considering a product and services catalog, sales orders and production orders management, among others.

This study case is conducted using the stable 3.5.2 version released on April 2014, and the implemented database is MySQL, with a size of 176 tables. The main method used to separate the words on this database is deleting the prefix `llx_` and separating the rest with underscore, but even with that, Dolibarr has a hyphenation accuracy of 92.63%, that is acceptable.

An important point in this ERP is that many words of the names of tables or columns are written in French, even when the main language of the database and the whole suite is clearly English. In order to study the incidence of a foreign language in the DB, all the words are ideally separated and counted, keeping a different register for French words; this can be seen in Table 4. The percentage of words in French on Dolibarr database is 11.38%.

In this study case, the procedure starts by distinguishing French words and build a list with their English meanings. After that, the French words are added as synonyms to the English counterpart, in their corresponding Synonyms Files. With this, the main KB is not modified, but GrACED acquires a very limited comprehension of French, enough to analyze Dolibarr.

From the pie chart of Fig. 12 it can be seen that 7.955% of the tables contains information related to the *Product Definition* category, while the remaining 92.045% does not. The bottom chart of Fig. 12 shows that 83.333% of the categorizations are Neutral, while the 11.111% is Tricky; the remaining 5.556% is of type True.

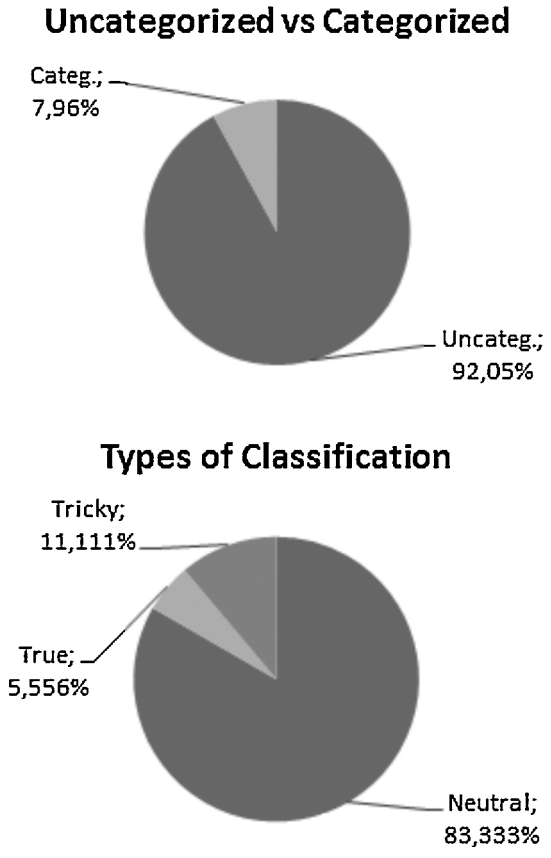

Fig. 12. Pie chart comparing categorized tables vs not categorized (top). Pie chart comparing types of categorizations (bottom). Dolibarr.

Like in the previous study case, the resulting classification done by the agent is compared to the manual results generated by a group of experts.

Once GrACED finishes its analysis, the outcome is a total of 14 categorizations, and only 1 of those is not previously manually classified. Still, when turning those results to the group of experts, they agree on the extra categorization, leaving 14 matches out of 16 classifications made by the experts; this results on a 87.50% of accuracy on GrACED's categorization of Dolibarr database.

This study case is deemed as successful, but it is important to note the difference the database size between OpenERP and Dolibarr, because the latter is 60% smaller in size. This derives on less tables containing information related to the Product Definition and thus less categorizations; due to having less classifications, each miss has a bigger impact on the resulting accuracy percentage.

4.3. Adempiere

The last study case is centered in Adempiere [30], another open-source ERP suite, published under a GNU (General Public License), as a *fork* of Compiere [24]. A *fork* happens when developers use the original source code of an open-source software package and start an independent development over this, creating different software, sometimes even with a new name. This is common in the open-source community.

This suite has a database of considerable size, with 726 tables and more than 14,000 columns, implemented in Oracle 10g XE. As a big difference from the other study cases, this database has a lower separation accuracy of 50.159% because there is no standardization in the use of a naming convention, even when the most recurrent in this suite is the underscore. Oracle is a case-sensitive DBMS, but all the names of this DB are written in uppercase, even when it was possible to use either Pascal or Camel casing.

Another issue on this database is the redundancy of tables that often store similar information: there are many duplicated data, hindering the maintenance and allowing the agent to over-classify this suite.

Due to the mentioned issues, and mostly the low separation accuracy, this suite is selected as a study case in order to check the behavior of the agent under non-optimal environments.

From these results, Fig. 13 (top) shows that 16.76% of the database is selected as containing *Product Definition* information; the remaining 83.24% of the DB does not contain manufacturing information. Also, the pie chart in Fig. 13 (bottom) displays the proportion of the types of Adempiere categorizations: 63.946% are Neutral classifications, while 17.007% are Tricky; the remaining 19.048% are of the True type.

As expected, the impact of redundancy and inconsistent hyphenation is reflected in coincidences between GrACED results and the manual classifications.

The agent finished with a result of 124 categorized tables, which consists of 93 classifications not previously acknowledged by the group of experts. However, from the feedback obtained from them, only 24 of the extra categories are acknowledged as valid, giving a total accuracy of 82.456% on GrACED's categorization of Adempiere's database.

In the presence of redundancy, lack of naming conventions and excessive use of generic words, GrACED tends to *over-categorize* the database. For example, a table marked as repeated three times (with slight modifications but storing the same information) is categorized only once by the experts but three times by the agent.

Nevertheless, this is a successful study case because it allowed sustaining the premises denoted at the beginning: there is a strong

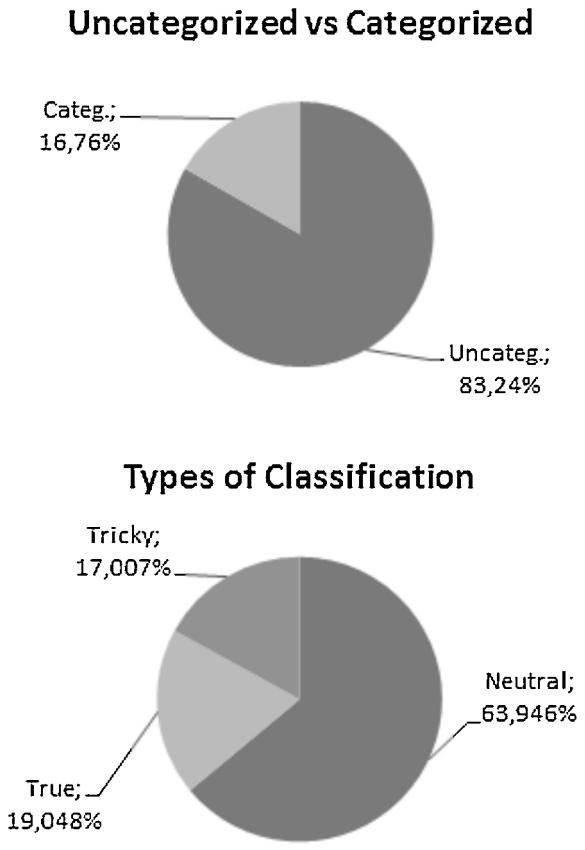

Fig. 13. Pie chart comparing categorized tables vs not categorized (top). Pie chart comparing types of categorizations (bottom). Adempiere.

dependence on the words separation and classification results while working with natural language. The use of generic words may impact on the categorization, deriving too many results to the Tricky type.

5. Conclusions

The current work proposes the basic structure for a knowledge-based intelligent agent named GrACED that works with natural language (English). GrACED uses a KB structured with weighted *bags-of-words* generated using the categories for manufacturing information proposed on the international standard ANSI/ISA-95.

GrACED works with an ERP database (the agent environment), and analyzes its content to study not only how the data is structured in tables, but also to find the needed information to integrate systems for various purposes related to the production administration. This agent also evaluates the distribution of the information in such categories in order to determine the matching of the ERP to handle production information.

This is especially useful to integrate the information systems of a supply chain's member, or to attempt to achieve collaboration between a business system and an APS (Advanced Planning and Scheduling) system. Another meaningful use is to permit a fast and simpler study of the current ERP system, to facilitate the extraction of the necessary information to link, for example, a mathematical model for production planning. This becomes a first step in order to facilitate the integration of the APS systems into the ERP's.

The agent functionality is evaluated through three study cases, always employing open-source ERP suites: OpenERP, Dolibarr and Adempiere, reaching a favorable behavior with accuracy higher than the 85% in the successful cases. However, the as a prototype that only implements one third of the full knowledge-base (the level-1 node of Product Definition), GrACED achieved good results.

This work presents several venues for future development, among them, implementing the complete graph derived from the ANSI/ISA-95 categorization, into the knowledge-base. In order to add more knowledge to the agent, more databases could be used to generate bigger and more diverse bags-pf-words. Regarding study cases, bigger and more diverse groups of experts should be included, in order to guarantee a wider range of experience and human-insight. Also, following this line, more study cases should be evaluated, even for databases that are not in English and whose words should be added to the Synonyms Files.

Acknowledgements

The authors gratefully acknowledge the financial support for the work presented in this article to CONICET through Project PIP 688, ANPCYT with grant PICT2012 2484 and Universidad Tecnológica Nacional through PID 25/O152.

References

[1] W. Shaojun, W. Gang, L. Min, G. Guoan, Enterprise resource planning implementation decision & optimization models, J. Syst. Eng. Electron. 19 (3) (2008) 513–521.
[2] M.A. Rothenberger, M. Srite, An investigation of customization in ERP system implementations, IEEE Trans. Eng. Manag. 56 (4) (2009) 663–676.
[3] H. Stadtler, Supply chain management and advanced planning – basics, overview and challenges, Eur. J. Oper. Res. 163 (3) (2005) 575–588.
[4] P. Hadaya, R. Pellerin, Determinants of advance planning and scheduling systems adoption, in: The Third International Conference on Software Engineering Advances, ICSEA, Sliema, 2008.
[5] A.J. Zoryk-Schalla, J.C. Fransoo, T.G. de Kok, Modeling the planning process in advanced planning systems, Inf. Manag. 42 (1) (2004) 75–87.
[6] J. Greenfield, K. Short, Software factories assembling applications with patterns, models, frameworks and tools, in: OOPSLA'03 Companion of the 18th Annual ACM SIGPLAN Conference on Object-oriented Programming, Systems, Languages, and Applications, New York, USA, 2003.
[7] R. Jardim-Goncalves, A. Grilo, A. Steiger-Galcao, Challenging the interoperability between computers in industry with MDA and SOA, Comput. Ind. 57 (8/9) (2006) 679–689.
[8] J. Browne, J. Zhang, Extended and virtual enterprises – similarities and difference, Int. J. Agile Manag. Syst. 1 (1) (1999) 30–36.
[9] D.E. Shobrys, D.C. White, Planning, scheduling and control systems: why cannot they work together, Comput. Chem. Eng. 26 (2002) 149–160.
[10] EU-Commission, MANUFACTURE – A Vision for 2020 Assuring the Future of Manufacturing in Europe, Office for Official Publications of the European Communities, 2004.
[11] E. Muñoz, E. Capón-García, A. Espuña, L. Puigjaner, Ontological framework for enterprise-wide integrated decision-making at operational level, Comput. Chem. Eng. 42 (2012) 217–234.
[12] ISA, ANSI/ISA-95.00.01-2000, Enterprise-Control System Integration. Part 1: Models and Terminology, 1st ed., 2000.
[13] L. Prades, F. Romero, A. Estruch, A. García-Domínguez, J. Serrano, Defining a methodology to design and implement business process models in BPMN according to the standard ANSI/ISA-95 in a manufacturing enterprise, in: The Manufacturing Engineering Society International Conference, MESIC, 2013.
[14] I. Harjunkoski, I. Nyström, A. Horch, Integration of scheduling and control – theory or practice? Comput. Chem. Eng. 33 (0098-1354) (2009) 1909–1918.
[15] D. He, A. Lobov, J.L. Martinez Lastra, ISA-95 tool for enterprise modeling, in: ICONS 2012, The Seventh International Conference on Systems, Saint Gilles, 2012.
[16] K. Nagorny, A.W. Colombo, U. Schmidtmann, A service- and multi-agent-oriented manufacturing automation architecture: an IEC 62264 level 2 compliant implementation, Comput. Ind. 63 (8) (2012) 813–823.
[17] G. Zhao, J. Deng, W. She, CLOVER: an agent-based approach to systems interoperability in cooperative design systems, Comput. Ind. 45 (3) (2001) 261–276.
[18] W. Shen, Q. Hao, H. Yoon, D.H. Norrie, Applications of agent-based systems in intelligent manufacturing: an updated review, Adv. Eng. Inform. 20 (4) (2006) 415–431.
[19] S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, 3rd ed., Prentice-Hall, 2009.
[20] H.M. Wallach, Topic modeling: beyond bag-of-words, in: Proceedings of the 23rd International Conference on Machine Learning, New York, USA, 2006.
[21] J. Roa, M. Pividori, M. Gutiérrez, G. Stegmayer, How to develop intelligent agents in an easy way with FAIA, in: F.V. Cipolla Ficarra (Ed.), Quality and Communicability for Interactive Hypermedia Systems: Concepts and Practices for Design, I. Gobal, 2010, 120–140.

[22] L. Brehm, A. Heinzl, M.L. Markus, Tailoring ERP systems: a spectrum of choices and their implications, in: Proceedings of the 34th Annual Hawaii International Conference on System Science, 2001.
[23] ISA, ANSI/ISA-95.00.03-2005, Enterprise-Control System Integration. Part 3: Activity Models of Manufacturing Operations Management, 3rd ed., 2005.
[24] A.L. Pretorious, Compiere 3, Packt Publishing Ltd., Birmingham, 2010.
[25] OpenERP S.A., OpenERP, 2012 Available: https://www.openerp.com/ (online; accessed 2014).
[26] Panorama Consulting Solutions, ERPNext, 2010 Available: http://panorama-consulting.com/erp-vendors/erpnext/ (online; accessed 2014).
[27] NightLabs Consulting GmbH, JFire, 2011 Available: http://www.jfire.net/ (online; accessed 2014).
[28] World Wide Web Consortium, 2nd ed., Extensible Markup Language (XML) 1.1, vol. 1, W3C, 2006.
[29] S. Butler, M. Wermelinger, Y. Yijun, H. Sharp, Relating identifier naming flaws and code quality: an empirical study, in: 16th Working Conference on Reverse Engineering (WCRE'09), Lille, 2009.
[30] B.C. Pamungkas, ADempiere 3.4 ERP Solutions, Packt Publishing, Birmingham, UK, 2009.
[31] L. Destailleur, Dolibar ERP/CRM, Dolibarr Foundation, 2014 Available: http://www.dolibarr.org/ (online; accessed 2014).
[32] D. Song, R.Y. Lau, P.D. Bruza, K.F. Wong, D.Y. Chen, An intelligent information agent for document title classification and filtering in document-intensive domains, Decis. Support Syst. 44 (1) (2007) 251–265.
[33] G. Louloudis, B. Gatos, I. Pratikakis, C. Halatsis, Text line and word segmentation of handwritten documents, Pattern Recognit. 42 (12) (2009) 3169–3183.
[34] M.J. Derek, Operand names influence operator precedence decisions, in: The ACCU Conference, 2008.
[35] B. Sigurd, M. Eeg-Olofsson, J. van de Weijer, Word length, sentence length and frequency – ZIPF revisited, Stud. Linguist. 58 (1) (2004) 37–52.
[36] B. New, L. Ferrand, C. Pallier, M. Brysbaert, Reexamining the word length effect in visual word recognition: new evidence from the English Lexicon Project, Psychon. Bull. Rev. 13 (1) (2006) 45–52.
[37] G.N.U. Affero, Affero General Public Licence, 2007 Available: http://www.gnu.org/licenses/agpl-3.0.html (online; accessed 20.04.14).
[38] Free Software Foundation Inc., GNU General Public Licence, 29 June 2007 Available: https://gnu.org/licenses/gpl.html (online; accessed 20.04.14).

Melina C. Vidoni is an Information System Engineer, from the Universidad Tecnológica Nacional, from Santa Fe, Argentina. She is a Ph.D. student in Information Systems at the Universidad Tecnológica Nacional Facultad Regional Santa Fe. She is currently a research fellow with a full-time scholarship under the supervision of Prof. Vecchietti at the National Council for Technical and Scientific Research of Argentina (CONICET) at INGAR (CONICET-UTN). Her main research interests are advanced planning systems, enterprise systems, and integration of advanced solving techniques in optimization of planning and scheduling in small-and-medium manufacturing enterprises.

Aldo R. Vecchietti is a Chemical Engineer from the Universidad Nacional del Litoral, Santa Fe, Argentina. He also obtained the Ph.D. in Chemical Engineering at the same University. He is an Independent Researcher of the National Council for Technical and Scientific Research of Argentina (CONICET) at INGAR (CONICET-Universidad Tecnológica Nacional). His expertise area is Process System Engineering, working on optimization mathematical models for planning and scheduling of manufacturing and production companies and its supply chain. He has an extensive experience in consulting works with private production companies. He is also Professor in the Information System Engineering Department at Universidad Tecnológica Nacional (Santa Fe, Argentina) and since 2013 is the Department Head.