Business Rules (BR); Expert Systems (KBS); LATC Algorithm.

A proposal for functional integration

Raul A. Raggio, Carlos A. Russell

SUMMARY: It is quite interesting to analyze the BRG manifesto and observe both the highlight of the role and the location of the logic surrounding decision management within an Organization. In general, since the introduction of Information Technology (IT) to support data processing, software applications ended up being the repository of said logic and owning it. In fact, the area with the opportunity to introduce a change within a part of this decision making scheme is the only one that can make it operational: the IT function. Up until now, IT has been the counterpart of the advantages of computerization: IT has enclosed the logic of the decisions surrounding those matters in a Black Box.

On the other hand, computer science has put emphasis on studying the problems and alternative solutions proposed to introduce in a computer the ability to select one action from other possible ones, which enables Organization management. To this very end, concepts have been introduced and developments have been made which, in general, fall within the domain of machine learning (ML) and, in particular, of Induction Machine Learning (IML). In this field, two main aspects can be recognized: the one derived from Data Mining and, especially from Big Data, which tries to extract knowledge from the enormous amount of data stored in its Databases. The other, obtained from Examples cases, being the most used for the construction of Expert Systems. Both make up a part of Artificial Intelligence.

The vision that the BRG has of the so-called Business Rules (BR) proposes a conception and characteristics that make it a panacea for those who have the natural responsibility, because of their roles, in business management of an organization. On the other hand, the fundamental question of artificial intelligence, which is Knowledge Acquisition, provides the accounts on the formalization achieved through its technological advances to consolidate that acquisition on solid foundations.

Both the BRG vision and the formalizations in knowledge management can reach a convergence that provides IT with more intelligent behaviors and results, operating under the formal conditions of the logic of decisions established and maintained in the Organization's business areas.

If achieved, it would be a way to alleviate the dependence that, in many cases, currently has on IT management and its technological platform to achieve changes in this logic,

To show that this is possible is the ultimate objective of this work.

Keywords: Business Rules (BR); Expert Systems (KBS); LATC Algorithm, Machine Learning by Examples; Management Objects; Smart Organizations.

1.INTRODUCCIÓN

2.1. Business Rules

The use of Business Rules (BR) is increasingly frequent to establish the flow of decisions in the management processes of the affairs of an Organization. Both BR (Business Rules) and ML (Machine Learning) use the "IF - THEN" format for the expression of their Rules, which was introduced by Newell and Simon as 'Production Rules' in 1972. [Young, Richard M (2)]. They also share a common difficulty: the "Knowledge Acquisition". Both topics have attracted the interest of many researchers in the field of artificial intelligence, since that first introduction by Newell and Simon.

We are going to introduce ourselves in some of the formal aspects of this way of representing knowledge in order to find, if possible, some solution to this difficulty. Newell and Simon proposed the 'Systems' or ' Production Rules', consisting of a collection of "IF - THEN" rules that, together, form a model of information processing of some cognitive task or range of tasks. The **Production Rules** have some special properties that make them very suitable for modeling cognition: the process of knowledge. Since its origin as a problem solving model, Production Systems have grown to become a great formalism for modeling cognitive skills and aspects of learning. - [Young, Richard M (2)]

That was the starting point for the development of the different modes of approaching the ways in which a machine can come to 'reason' as a human expert. (This should be understood as follows: when presenting a given problem, a computer arrives at a conclusion similar to that reached by a high-level human expert, in a specific field of knowledge; its 'Domain').

Research efforts, then, were focused on the problem of knowledge acquisition and that of learning processes, already started by Newell and Simon. As a result of these efforts, there was an accelerated development in the field of Machine Learning (ML), which ranged from Data Mining, to the present day, to Big Data and the semantic Web, all the way through fuzzy logic and neural networks.

The bias with which this situation was faced caused the impression of an apparent laxity in the formal development of what, among the different branches of AI, still today continues to be the one that produced the greatest development of cases, known as Expert Systems (ES) and, in particular, the 'Expert Rules-based Systems', also called 'Knowledge-Based Systems (KBS)'.

But the absence of formalities was not an obstacle to the development of important Expert Systems, which began with Dendral (a system oriented to chemical analysis and molecular structure). Working as a team, Feigenbaum, Buchanan and Lederberg developed the first successful knowledge-based system (KBS). The key to their success was mapping all the relevant theoretical knowledge, from its general form to highly specific rules (kind of like "cookbook recipes"). [Feigenbaum, E. Negnevitsky, M. 1971 (6)]

The next major project by Feigenbaum and others, also at Stanford University, was in the area of medical diagnosis. The project, called MYCIN, started in 1972. Later it became Edward Shortliffe's PhD thesis (Shortliffe, 1976). MYCIN was a Rules-Based Expert System for diagnosing infectious blood diseases.

Later, as early as 1993, Xindong Wu [3], from the 'Department of Artificial Intelligence, University of Edinburgh' states:

> *"Research on 'Machine Learning' has focused mainly on the induction of rules from sets of disordered examples, especially in 'Attribute Based Induction', a formalism where examples are described in terms of a fixed collection of attributes".*

At the same time, he places 'Knowledge Acquisition' as the main problem for the construction of Expert Systems, Feigenbaum, E. [4] one of the pioneers in the development of SE, also refers to this topic, pointing it out as "the 'Bottleneck" in the construction of Expert Systems".

2.1. Knowledge Acquisition

It is defined as: "The process of extracting, structuring and organizing the knowledge of a source, generally human experts, so that it can be used in software development, such as an ES.

This is often the main obstacle in the construction of an ES: "Obtaining a set of examples and describing them in the format of a Rule is considered to be the main obstacle to the construction of an ES. [Jones, P.H. (5)]

The dilemma that arises lies in an intrinsic weakness of inductive deduction, and that consists in the difficulty in verifying and validating its results, which could reach a number of incommensurable levels. In these terms, it has been proposed to settle the issue by attributing that responsibility to the group of experts in the domain: In this regard it is stated:

> *"The problem of automating inductive inference can be simplified by concentrating its purpose on the generation of the hypothesis, attributing to the human beings the question of how to properly validate and test them by known deductive and statistical inference methods."* [Michalski, R.S., (7)].

On the most appropriate way to acquire knowledge, Xindong Wu [3] reflects:

> *"Learning from the examples has been found to be not only a feasible way but also the only way to avoid the problem of the 'bottleneck' in knowledge. While it is often difficult for an expert to articulate his experience explicitly and clearly, it is generally relatively easy to document case studies of his ability on the job".*

That line of thought justifies that, both for those with knowledge of the domain on which the Rules will be established, and for the experts in Knowledge

Management, who should guide the steps in that acquisition, its integration into a single team is so important, in order to achieve that common goal.

2.1. Knowledge Representation

Jones, P.H. (5), on the Acquisition of knowledge, expresses:

> *"rules are generally easier to use for characterizing knowledge during 'knowledge acquisition'."*

> *"The initial knowledge base can be developed from written materials or from example cases described by the expert during early unstructured interviews. Initial rules should be treated as approximations and their wording should be general to avoid pressuring the expert. As additional cases are described during interviews, the rule base can be expanded."*

We have already seen that, from the works of Newell and Simon, the use of rules of the form "IF ... THEN" is advocated for the representation of acquired knowledge. In this regard, the latter cited also report the existence of problems caused by the selection of multiple 'actions', triggered by the satisfaction of the same 'condition', to the point of having to establish an item of "Conflict resolution" to solve these kinds of situations.", as expressed in [Jones, P.H. (5)].

2.1. Inductive Concept Learning

Acquiring knowledge from example cases is a clear example of 'inductive learning' and the expressions obtained from the example cases, generated from observation and experience, are known as 'inductive hypotheses'.

"Inductive learning" is a process of acquiring knowledge by inductive inferences from descriptions provided by experts. That process involves operations of generalization, specialization, transformation; correction and refinement of the representations of knowledge.

'Inductive concept learning' points to a type of inductive learning whose final products are symbolic descriptions expressed in terms and in high-level forms, oriented to human understanding.

The most frequently studied type is 'concept learning from examples' (also called 'concept acquisition') whose task is to induce general descriptions of concepts from specific cases of those concepts.

Although it is one of the most common ways of learning, it has a fundamental weakness: Except in special cases, the knowledge acquired cannot, in principle, be fully validated. They are hypotheses with a potentially infinite number of consequences, while only a finite number of confirmatory tests can be performed.

The problem of automating inductive inference can be simplified by concentrating on the purpose of hypothesis generation, attributing to humans the question of how to properly validate them".

Of the two aspects of inductive inference - the generation of plausible hypotheses and their validation, only the first is in the primary interest of inductive learning research. The problem of hypothesis validations is considered to be of minor (theoretical) importance, because it is assumed that the generated hypotheses will be judged by human experts, and tested by known deductive and statistical inference methods".

In cases of example learning, inference is the central operation. To explain them, a general paradigm for inductive inference is formulated. [Michalski, R.S., (7)]

2.1. Concept Acquisition

All the techniques to acquire knowledge face a significant difficulty that can be summarized as: "the certainty of taking into account all the possibilities of combining the variables that have been possible to identify" and in "the absence of inconsistencies between the different instances of a Rule "

They are considered essential conditions for a Rule (BR) to express Knowledge about a given domain. (Subset of the real world).

In practice, what is expressed means that, once the rules that express the knowledge acquired about a given process have been implemented, it is not possible that:

1. There is some case of reality subject to these Rules, to which no result can be assigned among all those established by it.
2. There is a case for which the Rule assigns more than one result and they differ from each other.

'Learning from Examples' is also called 'Concept Acquisition'.

Concept Acquisition produces descriptions for classifying objects into classes, based on the characteristics of the objects, their "Attributes". Michalski, R.S., [7]

In pursuit of formalizing the above this author proposes the following paradigm:

2.1. General paradigm for inductive inference.

In that case, the set of statements about what was observed or about the examples made by the experts: 'F' can be seen as a collection of implications:

$$F:\{\, e_{ik} \;\Rightarrow\; K_i \},i \;\in I$$

[3]

Where e_{ik} (*an example*) denotes the description of the example of the k^{th} example of the concept (class) affirmed by the predicate K_i (or classK_i) and 'I' is the set that enumerates K_i. *(It is assumed here that any given example represents only one concept).*

Inductive assertion H can be characterized as a **set of concept recognition rules**:

$$H:\{D_i \;\Rightarrow\; K_i \},i \;\in I$$

[4]

Where D_i denotes a description of the concept from class K_i, which means, D_i **it is an expression of conditions,** such that **when they are satisfied by an object, that object is**

considered an instance of the class K_i . By the definition of inductive assertion, we must have:

H |> F means that H specializes F). [5]

Substituting [3] and [4] for F and H, respectively, in [5] and with the appropriate transformations, the following conditions can be derived to be satisfied so that [5] holds:

$$\forall\, i \in I\ (E_i \Rightarrow D_i) \qquad [6]$$

$$\forall\, i,j \in I\ \big(D_i \Rightarrow \sim E_j\big), if\ j \neq i \qquad [7]$$

Where E(j), $i \in$ I is a description satisfied by all the example cases of the class K(*i*), and only for such examples. (The logical disjunction of the different example cases)

The expression [6] is called the Completeness Condition and [7] is the Consistency Condition. These two conditions are the requirements that must be met for an inductive assertion to be acceptable as a Rule of Concept recognition.

The Completeness and Consistency conditions provide the logical foundation of the algorithms for Learning Concepts from Examples.

(The total content of this point has been taken from [Michalski, R. (7)])

2.1. FORMAL FOUNDATIONS OBTAINED

1. A formal basis for analyzing a set of expressions of the form "IF…THEN".
2. The conditions to be met to consider this set as a Concept Recognition Rule.
3. The logical foundations for algorithm construction appropriate to that purpose.

(Note that the validity of the Results, Actions or Conclusions associated with each instance of the set of Rules (Conditional part), which are verified, verified and valid, is under the responsibility and control of the Group of Experts in the Domain, are outside the scope of these formal considerations).

2.1. RULES-BASED EXPERT SYSTEMS

As a presentation of Chapter 2 of his book on Artificial Intelligence, M. Negnevitsky [6; Chap.2] expresses verbatim:

> "In which the most popular option for building Knowledge-Based Systems is presented: Rules-based Expert Systems".

So we could consider that, if a set of Rules (BR) showed the qualities established by the Inductive Paradigm, they could become a basis for the construction of a Rules-based Expert System. A Knowledge Based System (KBS). According to the model proposed by Newell and Simon, which was followed for the construction of both the Dendral and the Mycin System at the University of Standford by the group led by Feigenbaum, already mentioned, a Rules-based Expert System has the following internal components: Knowledge Base, Case Database, Inference Engine, Explanation Facilities and User and Development and Maintenance interfaces (from the Knowledge Base and Explanation Facilities). Data systems are present as external components, as they currently operate in the Organization: Databases and Application Programs.

Figure 1 presents a complete diagram of its structure.

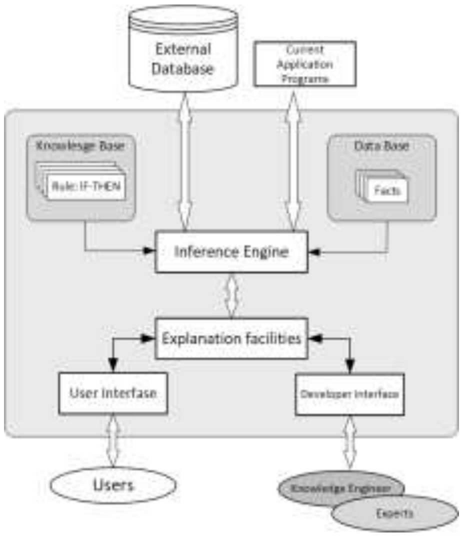
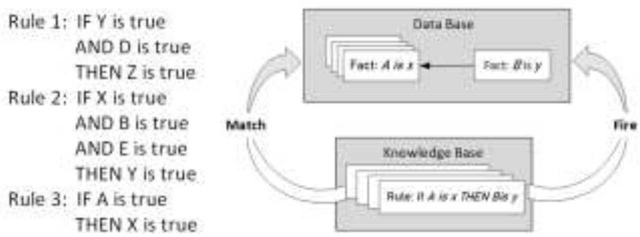


*Figure 1 – Inference Engine diagram. **

The Inference Engine, shown in Figure 1, works



schematically as follows:

*Figure 2 – The inference engine cycles via a match-fire procedure. **

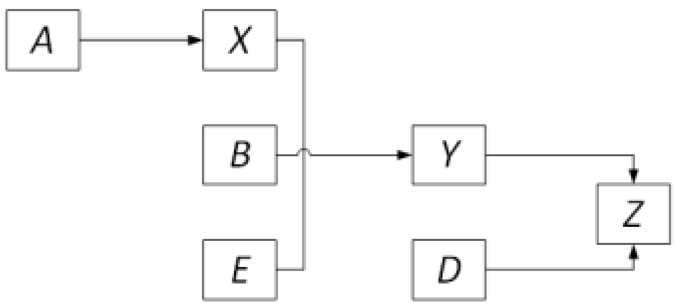


*Figure 3 – Example of an Inference chain. **

(*) All the figures have been taken from [Negnevitsky, M., (6)].

The chain of inferences shown in Figure 3 indicates how the expert system applies the rules to infer fact Z. First, Rule 3 is activated to deduce the new fact X from the given fact A. Then, Rule 2 is executed. to infer fact Y from initially known facts B and E, and fact already known X. And finally, Rule 1 applies the initially known fact D and the newly obtained fact Y to reach the conclusion Z. An expert system can show your chain of inference to explain how a particular conclusion was reached; This is an essential part of your "Explanatory Facilities" [Negnevitsky, M., (6)].

The schemes presented are inspired by the proposal by Newell and Simon to represent the observed pattern in the way in which human experts solved problems in relatively complex fields of knowledge. Perhaps that is why they called them 'Production Rules'.

From the observation of these schemes, it could be concluded that they are learning- oriented. But they are not. They produce 'facts', not knowledge.

Inference engines are geared towards answering questions such as, "Is it possible that V is true, based on the knowledge contained in your Base?" That is:

1) Can you infer V? and…

2) If so, can you show how you did it?

But learning is not about those issues.

More precisely, it is presented as one of three main deficiencies that this type of Expert Systems shows:

> ***Opaque relationships between rules***. *Although individual production rules tend to be relatively simple and self-documenting, their logical interactions within the broad set of rules can be opaque. Rule-based systems make it difficult to see how individual rules serve the overall strategy. This problem is related to the lack of hierarchical representation of knowledge in expert rule-based systems.*
>
> ***Ineffective search strategy***. *The inference engine applies an exhaustive search through all the production rules during each cycle. Expert systems with a large set of rules (over 100 rules) can be slow, and therefore large rule-based systems may not be suitable for real-time applications.*
>
> ***Inability to learn***. *In general, rule-based expert systems do not have the ability to learn from experience. Unlike a human expert, who knows when to "break the rules," an expert system cannot automatically modify its knowledge base, or adjust existing rules, or add new ones. The expert group remains responsible for reviewing and maintaining the system. From* [Negnevitsky, M. (6)].

Regarding the deficiencies raised, we will see later how the first one is solved, in the opportunity to face the functionality of a 'Repository'.

Regarding the second, having a set of Rules that satisfy the inductive paradigm, provides other alternatives for the pairing of cases and Rules, which do not require following the procedures used by the inference engines (Forward chaining and Backward Chaining).

And with regards to the inability to learn (automatically), it does not seem that it was a matter ignored 'a priori' as the verification of the results of application of the rules is outside the proper domain of the Expert System.

The most accepted way of acquiring new knowledge to develop Rule-based Expert Systems, is based on the 'Concept Acquisition from Examples', and expressing them in the' IF THEN 'form. Thus, automated assistance is only possible regarding the 'Conditional' part of the Rules.

2.1. Presentation of the LATC Algorithm

Taking into account the aforementioned deficiencies, and the conditions that a set of Rules must satisfy in order to be considered acceptable as a Concept recognition Rule, an algorithm called 'LATC' (Learning Assistant through Cardinality) has been developed, based on the Cardinality analysis of the Attribute Values contained in the expressions of the initial set of Examples cases, proposed by the 'Domain Expert Group'.

As a result, a complete and consistent set of the 'Conditional' part of the Rules is obtained, which satisfies the Inductive Paradigm. The Group of Experts, with the assistance of the experts in the administration of Knowledge, will associate the appropriate 'Actions' to the new instances obtained from the Rule.

This algorithm will be inserted into an Assistant with functions of record, development, verification, control and promotion of the Business Rules that have been decided to establish. These questions, as well as those derived from the performance environment in which the Assistant operates, will be the object of an analysis that will be dealt with in detail later on. The following is an example case of application of the 'LATC'.

The outline of a Business Rule (BR) considered in this document [1]

Id Rule	Rule's Name							
Instance	Conditions (Conditional part)						Actions	Explanatin
	At1	At2	...	Atj	...	Atn	Result or Derivation	
1	Ø $a_{(1),1}$	Ø $a_{(2),1}$	...	Ø $a_{(j),1}$	...	Ø $a_{(n),1}$	µ r_{1-g}	α (t)1
2	Ø $a_{(1),c}$	Ø $a_{(2),d}$	...	Ø $a_{(j),2}$	...	Ø $a_{(n),2}$	µ r_{3-b}	α (t)2
...	....	....	...	....	...	....	....	...
k	Ø $a_{(1),b}$	Ø $a_{(2),d}$	...	Ø $a_{(j),p}$	...	Ø $a_{(n),1}$	µ r_{1-g}	α (t)k
...	....	....	...	....	...	....	....	...
m	Ø $a_{(1),1}$	Ø $a_{(2),c}$	...	Ø $a_{(j),1}$	...	Ø $a_{(n),q}$	µ r_{1-g}	α (t)m

Figure 4 - Basic Outline of a Business Rule (BR)

Where:

- **Rule Id**: Unique Identification of a Rule (BR)
- **Instance**: Correlative number that identifies each of its instances. Each column of the Conditional Part of the Rule identifies an 'Attribute" and the cell formed by the Identifier number of an Instance and one of the Attributes columns, contains the expression of the Value that the Attribute assumes for that Instance.
- Associated with that Instance Identifier, in the Column of 'Actions' or 'Results' will be found the expression that indicates the consequence of having found all the values of the same Instance that coincide, one by one, with those of the attributes corresponding to those of the Case submitted to that Rule. That way of interpreting each row (Instance) of

[1] *In the diagrams of the Rules that will be presented, the components on which your attention is focused are shown, at each step of development. The complete scheme will be seen in the Repository.*

ISSN 2229-5518.

the Table that represents the expression of a Rule, is equivalent to the logical form:

IF $[\O_{(1,i)} \wedge \O_{(2,i)} \wedge \ldots \wedge \O_{(n,i)}]$ Then→ $\mu\ r_j$,

which is a logic predicate of order 1, without disjunction. *(When dealing with the Inference Engine, we will see the justification for this restriction on the use of disjunction)*

The format $\mu\ r_j$ for a Result, denotes with 'μ': Type of 'Result' : {Formula, Procedure, Rule} ; with 'r'; Unique identification within the type indicated by μ and finally with the subscript j , in r_j, an (eventual) option is identified within r. (A sub-type or a sub-Class of r).

- The expression $\alpha(t, i)$ denotes with α an Explanation Type: {Text or Pointer to a text); with (t, i), ,identifies the explanatory text of the Instance: (i), of the Rule (or Table), or the pointer to where it can be found.

The form of Table adopted in this document for the expression of a Rule (BR), where each row represents a Concept, is to facilitate a quick interpretation. This is the reason why, you should also try to restrict its number of Attributes to those that can be displayed on a single screen of a workstation, without having to 'navigate' it horizontally, to cover it in its entirety. Care must be taken to keep the expressions of the Rules (BR) within those limits.

On this question, Newell and Simon [8] state:

"All science characterizes the essential nature of the systems they study. These characterizations are invariably qualitative in nature, as they establish the terms within which more detailed knowledge can be developed. Its essence can often be captured in very short and very general statements. "

And Johnson, J, in his book - "Designing with the Mind in Mind" [9], states:

"*The principles of Gestalt visual perception, shows how our vision system is optimized to perceive structures ... when information is presented in a concise and structured way, it is easier for people to explore and understand it".*

Tanto lo mencionado anteriormente, como la restricción de no utilización de disyunciones en la expresión de cada instancia de un Regla (BR) tienen como finalidad evitar las deficiencias mencionadas por [Negnevitsky, M. (6)].

2.1. The initial Set of Examples or Cases.

The initial set of examples generated by the expert group to start the process of developing a Rule (BR) is required to:

1. Identify and represent all the pairs (Attributes, Value) that must be considered so that it is decidable to associate them with one of the Results or Actions specified in a Rule (BR).
2. Identify and Represent all the possible Results or Actions, within the domain on which you want to establish that Rule.

The set of Values belonging to the same Attribute, will determine the part of the Domain of the Rule, referred to that Attribute. The set of Domains of each of the attributes will determine the complete Domain of the Rule. No value outside the Domain will be considered by the Rule.

Note: *It would be desirable for the computer program that assists the Group of Experts in the task of building Rules, to have a 'parser' that controls the specified ranges, through a range of Values at an interval thereof. Although in common cases, a sheet of graph paper could be enough, where to draw the straight lines indicated by each of the intervals, taking scales, to control overlaps, unwanted gaps, or unbounded intervals.*

2. Looking at an Example.

2.1. Example Description.

Let us give a brief description of the Example that will be represented by Table 1, shown below. It is a very simplified version that represents the conditions to be satisfied to achieve the pension benefit for those persons who have carried out work in the country and made the planned social security contributions, within the framework of the legislation in force in the Argentine Republic. If such conditions are not satisfied, this would result in its denial.

Let us give a brief description of the Example that will be represented by Table 1, shown below. It is a very simplified version that represents the conditions to be satisfied to achieve the pension benefit for those persons who have carried out work in the country and made the planned social security contributions, within the framework of the legislation in force in the Argentine Republic. If such conditions are not satisfied, this would result in its denial.

2.2. Conditional Part of the Example Table

The number of Attributes do not differ from the real case, but the Values of those Attributes do. For this reason and for the benefit of its better understanding, in the selected example only the values of the attributes that turn out to be more significant and that do not require a very elaborate explanation of their meaning are considered:

***Sex**: Male or Female; **Activity Type**: I: Independent or D: Dependent (from an employer); **Contribution Years**: Number of years in which pension contributions have been made (integer number); **Age**: expressed in Years (Integer Number).*

2.3. 'Results' or 'Actions' from the Example.

As possible 'Results', three categories appear, namely: {YES, NO: Rv.}

The 'Rv' category corresponds to cases whose situation escapes the decision scheme expressed in the Rule: ('Table', hereinafter). By including this situation in the example, it is intended to indicate that the non-decision resolution of some cases, in certain circumstances, does not necessarily imply that the Rule cannot be established, following the criteria set forth by Pareto & Juran from '80: 20 '. - [Pareto, V. - Juran, J.M. (14)]

Having made this clarification, we are in a position to continue with the analysis object of this work.

2.4. Initial Sample Table

Initial Sample Table					
Instance #	Conditions				Actions
	Sex	Activity	Contribution (years)	Age	Result
1	M	D	≧ 30	≧ 65	YES
2	M	I	≧ 30	≧ 65	YES
3	F	D	≧ 30	≧ 60	YES
4	F	I	≧ 30	≧ 60	YES
5	F	I	≧ 30	< 60	NO
6	M	I	<30	≧ 65	YES
7	M	I	< 30	< 65	NO
8	F	D	<30	≧ 60	Rv

Table 1 – Initial sample table provided by the Expert Group

Analyzing the instances and the results that each implies, it can be said that each and every one is correct.

However, there is an Attribute that shows its related values by an 'order' relationship. That is the 'Age' attribute.

Its two possible values are {'≧ 60','≧ 65'}. They could overlap and, consequently, produce possible contradictions, in some cases. It would be appropriate to normalize these values.

(For reasons of clarity in the notation, we will not consider the upper or lower dimensions, in the expression of the Attribute Values expressed by intervals).

2.5. Normalized Initial Table

Once the pertinent adjustments are made, we will have a Normalized Initial Table, such as the one presented in Table 1a.

3. Cardinality, Completeness, Consistency and the 'LATC' Algorithm.

3.1. Application Example

With Table 1a normalized, we will carry out the Criteria checks and formal procedures proposed in the section 'Cardinality, Completeness and LATC Algorithm',

Instance	Conditions				Actions
	Sex	Activity	Contributions (years)	Age	Result
1	M	D	≧ 30	≧ 65	YES
2	M	I	≧ 30	≧ 65	YES
3	F	D	≧ 30	$60 \leq x < 65$	YES
4	F	I	≧ 30	$60 \leq x < 65$	YES
5	F	I	≧ 30	< 60	NO
6	M	I	<30	≧ 65	Rv
7	M	I	< 30	$60 \leq x < 65$	NO
8	F	D	<30	$60 \leq x < 65$	YES

which integrates this work.

Table 1a –Initial Table, with Normalized Domains

Notation:

We will note the Cardinality of $V(a_i)_j$ as '$Card\ (A)_i$ '[2], as a summary.

$Card[A_x, C_i^j]$: will note the Cardinality of Attribute A_x in Sub-Table C_i^j

$Card[(A_i)]_p$: $Card\ (A)_i$ in the instance *p* of selection, *p=0*, will note the initial instance.

Let's consider $V(a_i)_j = \bigcup_{j=1}^{n}\{(a_i)_j\}$

While $\{(a_i)_j\}$ is the value set of the '*n*' instances of each column from Table 1; that is: $V(a_i)_j$ is the name of the different value set that the Attribute Ai can assume in the '*n*' *(n=8)* instances of the Normalized Table 1.a. Therefore:

$V(a_1)_j = \bigcup_{j=1}^{n=18}\{(a1)_j\} = \{M; F\} \Rightarrow Card\ V(a_1)_j) = Card\ (A)_1 = 2.$

$V(a_2)_j = \bigcup_{j=1}^{n=18}\{(a2)_j\} = \{D; I\} \Rightarrow Card\ V(a_2)_j) = Card\ (A)_2 = 2.$

$V(a_3)_j = \bigcup_{j=1}^{n=18}\{(a3)_j\} = \{\geqq 30; > 30\} \Rightarrow Card\ V(a_3)_j = Card\ (A)_3 = 2.$

$V(a_4)_j = \bigcup_{j=1}^{n=18}\{(a4)_j\} = \{\geqq 65;\ 60 \leq x < 65; < 60\} \Rightarrow Card\ (A)_4 = 3.$

Extending the previous considerations to the 'Results'

ATTRIBUTE	Value$_1$	Freq$_1$	Value$_2$	Freq$_2$	Value$_3$	Freq$_3$	Cardinality
Sex	'M'	4	'F'	4	—	—	2
Activity	'D'	3	'I'	5	—	—	2
Contribution Yrs.	'≧ 30'	5	'< 30'	3	—	—	2
Age	'≧ 65'	3	$60 \leq x < 65$	4	'< 60'	1	3

Column, we will have:

$V(r)_j = \bigcup_{j=1}^{n=18}\{(r)_j\} = \{YES; NO; Rv\}; \Rightarrow Card\ V(r)_j = 3.$

Now, let's build the Frequency Table of C_i^0 as follows:

Table 2 – Frequencies - (Conditional Part)

The Frequency is obtained by adding the times that a value of an Attribute in Table 1 is referenced in each line (instance) of the Table. For example, the value '> = 65' (in gray in Table 1) appears 3 times.

Analogously it is calculated, with the total of the cases or instances contemplated and with the Results R_i from Table 1a, the Results Table of C_i^0.

	R_i			
Cases	Yes	No	Rv	Cardinality
8	5	2	1	3

Table 2a - Frecuencies (Results)

3.2. Cardinality Analysis of Tables Cp_i^p.

We will analyze this information using the formal foundations of the Appendix: "Cardinality, Completeness and LATC Algorithm", of this document, whose reading can be ignored.

In accordance with the Corollary **1** (*Theorem 1*),

If $q(T) < \prod_{i=1}^{n} Card\ (Ai) \Rightarrow$ 'T' is incomplete. **[9]**

Where $q(T)$ is the quantity of instances of a Table *'T'*.

In our case:

$q(C_i^0) = 18 < \prod_{i=1}^{n} Card\ (Ai) = \prod_{i=1}^{4} Card\ [A_i]_0$

*(Card A1*Card A2*Card A3*Card A4) = 2*2*2*3 = 24*

Therefore, C_i^0 is incomplete.

Applying Corollary 2**b** (*Theorem 2*), **[10b]**

[2] *"We will call the general concept that, by means of our active ability to think, arises from the set M when we abstract from nature and the order of its elements. We denote the result of this double act of abstraction by Card (M)* ". Georg Cantor, 1895.

(Because it is the Initial Table. That is: There are no previously selected attributes)

If $q\,C_i^j > 1$ y $q\,C_i^j < \prod_{i=1}^{n}[Card(A_i)]_0 \Rightarrow$

$\Rightarrow \exists A_x\ \varepsilon\,(C_i^j)$ such that *Card* $[A_x, C_i^j] <$ *Card* $[A_x, C_i^0]$;

Then, following the indications of the aforementioned Corollary, the Table Cp_i^p , will be constructed so that $\{C_i^p \cup Cp_i^p\}$ satisfies the ***Completeness Condition***

[0] 3.3. Construction of the Table $\boldsymbol{Cp_i^p}$ (LATC Algorithm)

Previous Steps

1. To construct Table Cp_i^0 we will firstly use, *(since there are no previously selected Attributes*), the result of the Corollary **3b** (*Theorem* 2) **[11b]**, which will indicate the value of $q(Cp_i^0)$;

   Therefore: $q(Cp_i^0) < \prod_{i=1}^{n}[Card(A_i)]_0$ -$q(C_i^0) = (24 - 8) = 16$

   That is: Cp_i^0 should consist of 16 instances.

2. Continuing with the construction steps of the Table Cp_i^0, we will use the Corollary **4** (*Theorem* 2) that shows us the format of the Table of <u>Contents of</u> Cp_i^p. By constructing it, the contents of the instances of the Table Cp_i^p**,** will be obtained, that is, the value of the Attributes that each will make up its instances. The form of the Table of Contents Cp_i^p , follows:

Table 3 – Contents of $\boldsymbol{Cp_i^p}$ form

3. Corollary **4** (*Theorem* 2), **[12]**, provides us with the way to obtain the result of the Attribute Values to be considered in Cp_i^0 . Indicates that the following formula must be applied to obtain these values:

$$\forall y,\ k(ax)_y = \frac{Card\ (C_i^0)}{Card\ [(Ax),C_i^0]} - Frec\ [(ax)_y, C_i^p] \qquad \textbf{[12]}$$

4. We start with the first value, the first Attribute: *M*.

   For $k(Sex = \text{'}M\text{'}) = \frac{24}{2} - 4 = 8;$ that is:

   The number of instances that the TableCp_i^0 must contemplate for '*Sex* = 'M'' is '8'. Therefore:

   For $k(Sex = \text{'}F\text{'}) = \frac{24}{2} - 4 = 8;$ so we will have:

Contents of Cp_i^0	Attribute	Missing Instances	Attribute Values to consider	
Cardinality of C_i^p *in completeness*	'Sex'	16	'M'	8
Card $[Cp_i^p]_n = 24$			'F'	8

Table 4 - Contents of $\boldsymbol{Cp_i^p}$ for 'Sex' Attribute

'*Activity*', '*Contribution Yrs.*' and '*Age*' are the Attributes that have not been selected so far, yet. We will continue forming the Table started in 1), adding those Attributes after the one already established. Therefore, the Table of Contents for Cp_i^p will be as follows:

Table 5 - Contents of $\boldsymbol{Cp_i^p}$ pattern for the all attributes of Table 2

Reapplying the formula **[12]** used in point 3, we will successively obtain the values corresponding to the Attributes not yet considered:

For $k(Activity = \text{'}D\text{'}) = \frac{24}{2} - 3 = 9;$ which means:

For the D value of the *'Activity'* Attribute, we will have to consider 9 instances of Cp_i^p .

For $k(Activity = \text{'}I\text{'}) = \frac{24}{2} - 5 = 7;$ which

Contents of Cp_i^p ***(Complete)***	Attributes	Missing Instances (*)	Attribute Values to consider	
Cardinality of C_i^p *in completeness*	'Sex'	$\sum_{y=1}^{z} k(ax)_y$	'M'	#
Card $[C_i^p]_n = 24$			'F'	#
	'Activity'		$(ax)_1$	$k(ax)_1$
			—	---
			$(ax)_x$	$k(ax)_x$
	'Contribution Yrs.'		$(ax)_1$	$k(ay)_1$
			—	---
			$(ax)_x$	$k(ay)_x$
	'Age'		$(aw)_1$	$k(aw)_1$
			$(aw)_w$	$k(az)_x$

means:

For the *I* value of the *'Activity'* Attribute, we will have to consider 7 instances of Cp_i^p.

$k(Contribution\ Yrs. \geq 30) = \frac{24}{2} - 5 = 7;$

For $k(Contribution\ Yrs. < 30) = \frac{24}{2} - 3 = 9;$

For $k(Age \geq \text{'}65\text{'}) = \frac{24}{3} - 3 = 5;$

For $k(Age;\ 60 \leq x < 65\ ) = \frac{24}{3} - 4 = 4$

For $k(Age < \text{'}60\text{'}) = \frac{24}{3} - 1 = 7$

5. The Table of *Contents of* Cp_i^p will be completed as follows:

Contents of Cp_i^p ***(Qty.)***	Attributes not selected	Missing Instances	Values to consider	
			Value	*Quantity*
Cardinality of C_i^p *in completeness*	'Sex'	6	'M'	8
Card $[C_i^p]_n = 24$			'F'	8
	'Activity'		'D'	9
			'I'	7
	'Contribution Yrs.'		'≧ 30'	7
			'< 30'	9
	'Age'		'≧ 65'	5
			'60 ≤ x < 65'	4
			'< 60'	7

Table 6 - Contents of $\boldsymbol{Cp_i^p}$ completed

3.4. LATC Algorithm Application

Construction Procedure for Table $\boldsymbol{Cp_i^p}$**.**

Let's recall the values of the Frequency Table of C_i^0: *(Table 2)*

ATTRIBUTE	Value 1	Freq1	Value2	Freq2	Value3	Freq3	Cardinality
Sex	'M'	4	'F'	4	—	—	2
Activity	'D'	3	'I'	5	—	—	2
Contribution Yrs.	'≧ 30'	5	'< 30'	3	—	—	2
Age	'≧ 65'	3	60 ≤ x < 65	4	'< 60'	1	3

Following the sequence indicated in Section **'Construction of Table** Cp_i^p **– LATC Algorithm'** of the Appendix : **'Cardinality, Completeness and LATC**

Sum of frequencies of $\{C_i^0 \cup Cp_i^0\}$							
ATTRIBUTE	Value 1	Freq1	Value2	Freq2	Value3	Freq3	Cardinality
Sex	'M'	12	'F'	12	—	—	2
Activity	'D'	12	'I'	12	—	—	2
Contribution Yrs.	'≧ 30'	12	'< 30'	12	—	—	2
Age	'≧ 65'	8	60 ≤ x < 65	8	'< 60'	8	3

Algorithm', of this document.

Table 7 - Sum of frequencies of $\{C_i^0 \cup Cp_i^0\}$

It remains to present to the Group of Experts, the Complementary table Cp_i^p, in order to assign the relevant Result Value (Table *B Bis*), belonging to the set $\{R\}_i$.

The content of Cp_i^p is shown below:

Cp_i^p	Result of Complementary Rule of the Example					Valorization
	Conditions				Actions	Actions
#	Sex	Activity	Contribution Yrs.	Age	Results	Results
1	M	'D'	'≥ 30'	'< 60'	¿?	NO
2	M	'D'	'≥ 30'	60 ≤ x < 65	¿?	NO
4	M	'D'	'< 30'	'< 60'	¿?	NO
5	M	'D'	'< 30'	60 ≤ x < 65	¿?	NO
6	M	'D'	'< 30'	'≥ 65'	¿?	YES
7	M	'I'	'≥ 30'	'< 60'	¿?	NO
8	M	'I'	'≥ 30'	60 ≤ x < 65	¿?	NO
11	M	'I'	'< 30'	'< 60'	¿?	NO
13	F	'D'	'≥ 30'	'< 60'	¿?	NO
15	F	'D'	'≥ 30'	'≥ 65'	¿?	YES
16	F	'D'	'< 30'	'< 60'	¿?	NO
18	F	'D'	'< 30'	'≥ 65'	¿?	YES
21	F	'I'	'≥ 30'	'≥ 65'	¿?	YES
22	F	'I'	'< 30'	'< 60'	¿?	NO
23	F	'I'	'< 30'	60 ≤ x < 65	¿?	Rv
24	F	'I'	'< 30'	'≥ 65'	¿?	Rv

Table 8 *Table 8 Bis*

Once we've updated *Table 1 Normalized*, with the 'Conditional' instances obtained by the application of 'LATC' (*Table 8*) and the results provided by the Experts (*Table 8 Bis*), we get Table 9 and 9 Bis. *(Appears between […], the Instance # of the Initial Table, after the Value of its Result).*

	#	'm' cols. (j) Sex	Activity	Contribution Yrs.	Age	Results	Final Results
Rows n Index i	1	M	'D'	'≥ 30'	'< 60'	¿?	NO
	2	M	'D'	'≥ 30'	60 ≤ x < 65	¿?	NO
	3	M	'D'	'≥ 30'	'≥ 65'	YES [1]	YES [1]
	4	M	'D'	'< 30'	'< 60'	¿?	NO
	5	M	'D'	'< 30'	60 ≤ x < 65	¿?	NO
	6	M	'D'	'< 30'	'≥ 65'	¿?	[illegible]
	7	M	'I'	'≥ 30'	'< 60'	¿?	NO
	8	M	'I'	'≥ 30'	60 ≤ x < 65	¿?	NO
	9	M	'I'	'≥ 30'	'≥ 65'	YES [2]	YES [2]
	10	M	'I'	'< 30'	'≥ 65'	Rv [6]	Rv [6]
	11	M	'I'	'< 30'	'< 60'	¿?	NO
	12	M	'I'	'< 30'	60 ≤ x < 65	NO [7]	NO [7]
	13	F	'D'	'≥ 30'	'< 60'	¿?	NO
	14	F	'D'	'≥ 30'	60 ≤ x < 65	YES [3]	YES [3]
	15	F	'D'	'≥ 30'	'≥ 65'	¿?	YES
	16	F	'D'	'< 30'	'< 60'	¿?	NO
	17	F	'D'	'< 30'	60 ≤ x < 65	YES [8]	YES
	18	F	'D'	'< 30'	'≥ 65'	¿?	YES
	19	F	'I'	'≥ 30'	'< 60'	NO [5]	NO [5]
	20	F	'I'	'≥ 30'	60 ≤ x < 65	YES [4]	YES [4]
	21	F	'I'	'≥ 30'	'≥ 65'	¿?	YES
	22	F	'I'	'< 30'	'< 60'	¿?	NO
	23	F	'I'	'< 30'	60 ≤ x < 65	¿?	Rv
	24	F	'I'	'< 30'	'≥ 65'	¿?	Rv
	from 'LATC' Algorithm						From Experts Group

Table 9 *Table 9 Bis*

Then it can be verified that 'Conditional Part' of Table $\{C_i^p \cup Cp_i^p\}$, obtained by application of the LATC algorithm, satisfies Theorem 1:

$$\text{if } `T' \textit{ is complete} \;\Rightarrow\; (T) = \prod_{i=1}^{n} Card\,(A_i). \quad [8]$$

Note: *Note that, by construction, there are not two instances of the Table obtained whose 'Conditional' parts coincide.*

3.5. Conclusions at the End of the Example

Consistency

Due to the construction process applied and as can be seen in Table B, there are no possibilities of finding in the 'conditional part' of the Rule, two or more instances that turn out to be the same.

Therefore, the Rule obtained as a result of the LATC algorithm would satisfy the Consistency Condition, regarding the Conditional Part of that rule.

When the Group of Experts assigns the Results corresponding to each one of the proposed instances, the complete rule will be Consistent.

Completeness

It has already been seen in point [8] that the Completeness Condition has been reached as a result of the application of the Algorithm.

Therefore, it is possible to conclude that, by applying the algorithm to an initial set of example cases, presented by the group of experts and, having completed the Results corresponding to each of the new instances generated by the algorithm. In this way, the resulting Table satisfies the Inductive Paradigm proposed by Michalski [7].

4. Rule (BR)Operation

Once the Rule (BR): [A] has been obtained, so that it satisfies its completeness and consistency, its operation is simple.

Two modes are proposed for this;

4.1. Prime Mode:

1. The operation requires substituting, in the working memory, each one of the Values of the Rule [A] by respective prime numbers so that none of those are reused, more than for a single Attribute Value in [A].

   So, if $a_{(i,j)\in}\,[A] \;\Rightarrow\; for\ each\ i_1^n\ P_{i\,=} \prod_{j=1}^{j=m} a_{1,j}$. The same is done with the Values $\{v_j\}$ of the Attributes of $case_x$ to process; getting K_j; *(being* $1 \le i \le n; 1 \le j \le m$).

2. So, the Result of applying to $case_{\,x}$ will be the one assigned to R_i , so that $R_{i\,=}\,P_i$. Also associated is Texts T_i , explanatory of the result R_i. Both will be available for deployment, which will be referred to later. (*The Fundamental Theorem of Arithmetic ensure the uniqueness of each* P_i *and* K_i).

As it is a product of prime numbers, in certain circumstances the result can reach a significant number in digits. That is why the 'Overflow' or, that it does not truncate the maximum of significant figures in the obtained result, must be controlled. If it occurs, you can choose SHA Mode.

4.2. SHA MODE: [3]

1. In this mode, for each i_1^n, the array formed by $\{a_{(i,1)}; a_{(i,2)};;;; a_{(i,m)}\}$ is taken as a single string of characters such that $Cad_i\{a_{(i,1)}; a_{(i,2)};;;; a_{(i,m)}\} \Rightarrow$
$\Rightarrow P_i$ = SHA512 $\{\{a_{(i,1)}; a_{(i,2)};;;; a_{(i,m)}\}\}$.
2. Similarly, the chain composed of the Attribute Values of the $case_x$ and you get K_j in SHA mode. And it continues operating as indicated in point 2.) from Prime Mode. In this case, what must be controlled is that there are no duplicates obtained in P_i. Otherwise, it will be necessary to replace, in one of the P_i that duplicates are found, one of the Attribute Values of the row corresponding to the P_i selected and the operation is repeated from point 1 of the SHA Mode iterating until no longer produce duplications. in the $P_i, f\ or\ all\ p\ (i)\ such\ that\ 1 \leq i \leq m$.

With the operation of the Rule (BR) in any of the presented modes, the Rule Engine or Inference Engine is replaced and, with it, any possible error due to the transcription of the terms in which Rule [A] has been conceived.

What was and can be seen from the content of the Rule, is what will be executed in Production.

The promotion of a Rule (BR), from the end of its testing and set-up, to deployment in production, can use a 'Digital Signature' in accordance to the level of Authorization and the level of Control that the Organization considers pertinent for it.

With the digital signature, not only will its content be authenticated, but it will be protected from any subsequent unauthorized alteration.

If it should be necessary for a Rule Tables (BR) to have the possibility of executing more than one alternative version of the Rule, selecting it, for example, for its Validity, it is a situation that would be feasible, provided that the way exists, (For example: establishing another preceding Execution Rule), which determines the version of Rule (BR) to be executed, for a given case.

Having this alternative would allow greater flexibility in its use.

[3] **SHA** - is a set of cryptographic HASH functions (SHA-224, SHA--256, SHA-384, SHA-512) designed by the National Security Agency (NSA) and published in 2001 by the National Institute of Standards and Technology (NIST) as a Federal Information Processing Standard (FIPS) in the USA,

HASH - Mathematical functions that summarize the number of data characters in the Domain of the Function set.

4.3. ASSISTANCE SOFTWARE AND RULE (BR) VERIFICATION

In addition to giving support to all the activities that, on a Rule (BR), have been reviewed so far, the Assistance Software must also facilitate a set of activities that allow it to continue developing it.

Once the expert group has completed the Results of the instances added by the LATC algorithm, the experimental verification stage can begin. For this, the group of experts must have the facility contemplated in the assistance software, which allows them to enter the values of the variables necessary to execute The Rule in operation mode.

This facility must provide the display of a list of the Attributes contemplated in the Rule and a sort of grid, under the name of each Attribute, where they can insert the tuple of values they want to test, or select those values from the lists of the same ones that, optionally, have to be deployed.

Each tuple will represent the set of values of the Attributes contemplated in the Rule, which will represent an example case to be tested, imaginary or real.

The quantity, distribution and aesthetics of that presentation screen is a matter of implementation. (It is irrelevant to deal with that point in this document).

What does matter is that the operation is iterated as many times as necessary until the expert group considers that the behavior of the Rule is as expected. Each rank, tested; each Result achieved. Those circumstances can be recorded in a 'log'.

The validation of the established operating ranges for each Attribute is an important matter. The case of finding any plausible Value that is outside that range, is a situation that has already been contemplated, but this is the time to put it to the test.

Therefore, when verifying the Rule, it would be appropriate to select its Values from among those displayed as determining the Domain of each Attribute and, as a whole, of the Rule Operation Domain.

We have already mentioned that the last instance of the Table would be added to the one obtained as a result of the LATC and would correspond to the "Out of Range", with its Value equal to 'Null' for the Attribute in that instance of the Rule, which would be out of the established range. That last row of the Table, added specifically for that reason, could be assumed by the assistant program itself.

Another component that must be verified is the text associated with the instance that is selected by the Rule. Opportunity to check if your content is explicit and adjust it, if necessary.

At this stage of the analysis, we can conclude that all the data involved in the Table that represents the Rule, should be registered and protected. This question concerns the 'Repository' which, for now, it would be enough to consider as a "Database that registers information on the data used in the Rules (BR): (metadata)" and other elements that will be seen later.

4.4. Rule Analysis

At the conclusion of this verification stage, our attention should be drawn to how a Rule would operate in the reality of operational management. This is: in Production.

In this area, the 'identity' of the person requesting the service, (*situation for which the Example Rule will be executed*) will correspond to the other data referred to as 'Attributes' in that Rule: {Sex; Age: Years of Contribution: Dependent or Autonomous}. Their 'values' will be those corresponding to the identity of the Person requesting it.

That such 'Person' is the name assigned to an 'Entity', in which the Attributes that are assigned to a thing or object in the real world are grouped, to be used in all Expressions that are made about it and that, so far they are: {Identity; Sex; Age}. While {Years of Contribution and Cat: {*Ind* or *Dep*}} do not depend only on the identity of the Person, but on a 'Relationship' between 'Person and Company' (in the case of being a dependent job

(Cat = *Dep*)} or (Person = *Company*) in the case of self-employment: (Cat = *Ind*) This relationship may not be unique in a period of 30 years, considered for your right to the requested service.

So now we have more data to consider, and more data on that new data. Thus two new 'Entities' appear: 'Contributions' and 'Companies' and a new component: 'Relationship'. 'Contributions'; arises from a 'Relationship' between 'Person' and 'Companies'. This is more knowledge that we obtain from the analysis that we are carrying out on the Rule and it is important to be able to record it. This also concerns and must be available in the 'Repository'.

The time has come to join the team of experts, the experts in Computer Systems that act to support the data management of the Organization's operations.

It must be possible to verify that the operation of the Rule (BR) is possible. The ideal profile to integrate the team of experts is that of the Database Administration.

(It could be thought that, in the absence of such data, they could well be incorporated by whoever operates the Rule. Even though it would be technically possible, we must bear in mind that a high degree of reliability is required on this type of information and that they may have elapsed more than thirty years since its registration).

The participation of the experts in the Data Systems really matter for the following reasons:

- The first is to know if these data could be available. If there really is a record of them and if they are in a format that is adjusted to the needs of the Rule. Otherwise, input would be necessary to determine an appropriate way to obtain them.
- The second is to know if it is possible to access this information from the environment in which the Rule (BR) is expected to operate, in order to recover them or, even better (for various reasons, including security) if it would be possible to build an interface between the different operating environments.
- The third is the collaboration in the development of the specifications for the recovery of this information, all of which should be registered in the Repository, with the appropriate registration format for it.
- The fourth is to analyze whether it would be possible to articulate the functions of the current Data System, with the results of the application of the Rule and, in such a case, to establish the information exchange format to make it operational. That is: to extend the functional scope of the Interface. It would be an important objective to achieve: Neither more nor less than the integration of the Rules, now in the hands of those responsible for the relevant Business Area, with the Application Systems that support the Organization's data operations.

The Analysis stage of the verification of a Rule will be finalized when, for all the data required by it, it is reliably determined that each of them will be available at the time of operation of that Rule in Production.

The conformation of the specific registrants planned for the Repository will be an excellent specification platform for Data Systems. As an example, let's look at the following case.

4.5. The case of the 'Age (Person)' Attribute

The 'Age' Attribute, associated with the 'Person' Entity, is a case of no non-lasting value. If we consider its expression format: '*yyyymmdd*', we will conclude that each day that passes, the value of its content will vary. On the other hand, if the process of your benefit begins before reaching the minimum value of your age, the computable age will not be the distance from your date of birth to the day your process begins. In case something else is missing, the validity of the benefit would also depend on having the required certifications on your 'Contributions', to demonstrate your right.

The 'Effective Date of the Benefit' will then be that corresponding to the 'Minimum Age for the Benefit' of the person requesting it, if their certifications are complete or, otherwise, the Date on which they conclude their required certifications to be granted, which may be later than the date mentioned above, as indicated by the situations represented in the Example Table.

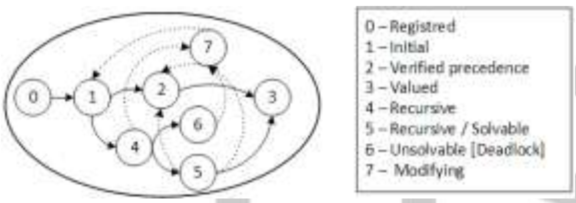
Therefore, the Attribute Value 'Age'; will be the result of a computation between the 'Date of Birth (Person)' and the 'Minimum Age for the benefit'. And that will be the one that determines the 'Validity' of the person's right to access retirement funds, subject to the condition of having completed the certification of their 'Contribution Years'. Hence, the need arises to establish a Rule.

To determine if it will be necessary to build a Rule for the valorization of the attribute 'Age (Person)', it will be sufficient to consider whether or not to proceed is subject to 'Conditions'.

The 'Minimum age for the benefit ', whose Values appear as dimensions in the ranges of the 'Age' Attribute, are considered 'Parameters' of the "Domain" to which the Rule belongs and, as can be seen in it, the Choice Value depends on the 'Sex' Attribute. If the Parameters are specialized by 'Sex', then to determine the value of the 'Minimum age …' a Formula will suffice.

The presentation of the 'Benefit Request'[4] it is the event that triggers the initiation of a 'Procedure', in which the events that occurred during the management of that Request will be recorded. An Attribute is 'Start Date'.

We will now see how this new knowledge is registered in the Repository and the utility it provides. To recognize the situation of each component, which may be evolutionary, a 'State' will be associated to each one of the components of the Repository. This evolution is



represented in the following diagram:

Figure 5 - Network of States for Repository Components
(Dotted lines: alternative paths in cases of modification)

5. Repository

5.1. Registration

Here is the example Rule (BR), registered in the Repository at the stage of the analysis that we are carrying out. This record is complete for the case of the examples, which has been formed in this way, for reasons of space. An image of that record will be transferred to the working memory in which the group of experts operates and on that image it will be updated. As it can be seen, all the States associated to each one of the Attributes whose 'State' are in '0', indicates that they are only registered in the format of the Conditional part of a Rule (BR) in the Repository.

Now, it remains for us to incorporate the new knowledge acquired. part of a Rule (BR) in the Repository.

Now, it remains for us to incorporate the new

#	COND	Id Regla Y	Derecho al Beneficio					
#	Sex (Person)	[illegible] (Person)	[illegible] (Person)	Edad (Person)	Resultados		ACCION	Textos Explicativos
1	M	D	20	≥ 60	T	ST	Procs / Título Proc	T [illegible]
2	M	I	20	≥ 63	T	ST	Procs / Título Proc	T [illegible]
3	M	D	20	60< x <63	T	ST	Procs / Título Proc	T [illegible]
4	F	I	20	60< x <63	T	ST	Procs / Título Proc	T [illegible]
5	F	I	20	x < 60	T	NO	Procs / Título Proc	T Motivo del Rechazo
6	M	I	>=30	61	T	Rc	Procs / Título Proc	T Pers y Trámite Completo
7	M	I	>=30	40 ≤ x <63	T	NO	Procs / Título Proc	T Motivo del Rechazo
8	F	D	x < 30	60 ≤ x <63	T	ST	Procs / Título Proc	T [illegible]

only possible to clarify that it can be included within the Rules system. See Chap. "Interfaces entre el Objeto y su Contexto" in Raggio R.A. [10].

knowledge acquired.

Figure 6 - Registration - Conditional part of the 'Procedure' Rule

All the dimensions in the intervals that value the different instances of 'Age (Person)' and 'Contributions (Person)' will be cataloged as: '*Vp*', indicated in 'red'. The Rule Status equals '0', in 'red '.

We will still need to record:

'Effective date'; 'Benefit Request'; 'Procedure'.

The 'Formula' that establishes the "Age (Person)" at the beginning of the right to the Benefit.

The rule that establishes the value of the Attribute 'Validity '.

To do this, we begin by registering, in an image for the Repository, a new Entity that we will call 'Procedure', using the main 'patterns' [5] which, for this purpose, are protected in the Repository.

The Attribute 'Initiation Date' is registered in the new Entity 'Procedure'. Also 'Effective Date'.

Figure 7 - Attribute Registration in the Entity 'Procedure'.

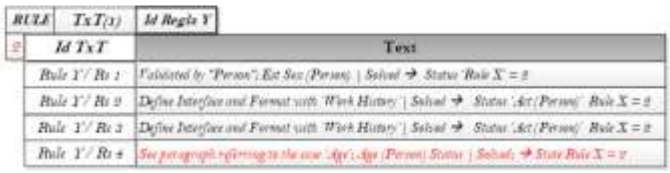
The 'Procedure ID' is the Attribute of univocal identification of that Entity in the repository. It is analogous to the primary Key of a Relational Database. The formats corresponding to each of the attributes are also recorded. Its 'Relationships' indicate that it is related to 'Rule X' and the Entity 'Person'.

Note: *the gray part of the Entity Scheme 'Procedure', indicative of the aggregates made in its registry, as the analysis progresses to determine the origin of each Value, necessary for the execution of the Rule in Production time.*

'Contribution Years' has not been registered, because the Method to obtain its Value has not yet been defined. With this composition of Attributes, the Entity 'Procedure' observes the 'Status' = 2.

In order not to complicate the example further, we will assume that we are dealing with cases in which the 'Contributions' were made in the same 'Company', with no continuity

solution. In such case, the Entity 'Procedure' will show the addition in Red in its registry for the Repository.



RULE	TxT(1)	Id Regla Y
2	Id TxT	Text
	Rule Y / Rt 1	Validated by "Person": Ext Sex (Person) | Solved → Status 'Rule X' = 2
	Rule Y / Rt 2	Define Interface and Format with 'Work History' | Solved → Status 'Act (Person)' Rule X = 2
	Rule Y / Rt 3	Define Interface and Format with 'Work History' | Solved → Status 'Act (Person)' Rule X = 2
	Rule Y / Rt 4	See paragraph referring to the case 'Age': Age (Person) Status | Solved; → State Rule X = 2

Figure 8- Registration of the 'Action' part of Rule Y.
(Having resolved 'ContributionsYears')

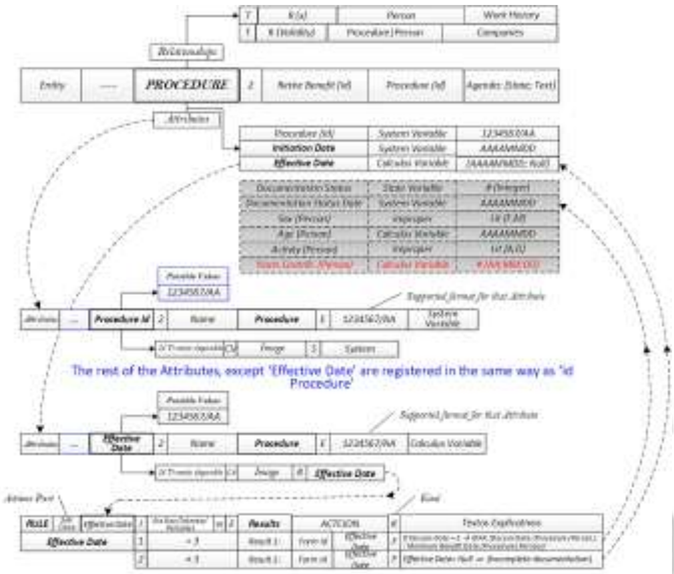
[5] **Patterns**: They are the models of the different Registers that can be contained in the Repository.

Now we just need to incorporate the "Texts" part of 'Rule Y':

Figure 9 - Registration of the 'Text' part of Rule Y.

Figure 10 - Flow of Activities and Registration of the Attribute: 'Effective Date'.

After completing this part, the State of the Conditional

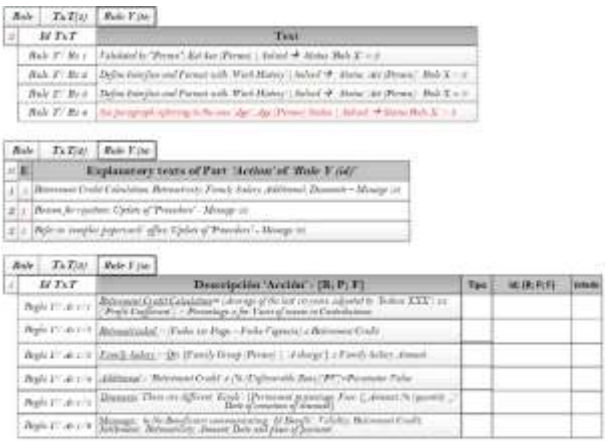


Part of Rule Y becomes 2, given that the State of all the

Figure 10 -Flow of activities on the Attribute Effective Date

Attributes mentioned in said part of the Rule, also reached, at the time. the State = 2.

This concludes the Verification Cycle of the 'Conditional' part of the Rule. **[1]**



Now it remains to analyze the part corresponding to the "Action" part of the Rule.

The ones shown in the Figure are the components that need to be developed to complete the analysis of the operation of a Rule at Production time.

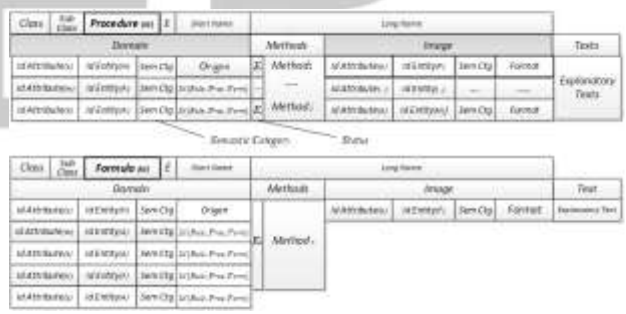
In this regard Jones, P.H. [5] indicates:

"The initial rules should be treated as approximations and their wording should be general to avoid putting pressure on the expert. As additional cases are described during interviews, the rule base can be expanded. Once a stable rule base begins to develop, you can provide feedback to structure interviews. Initially, rules and procedures can be manually traced with the expert considering each step. The same pattern of crawling through rules should continue once a version of the knowledge base is developed on a computer and its frequent use should be part of the process." Jones, P.H. [5].

Figure 11 - Progression of descriptive and explanatory texts.

The progression of these types of Texts will be given as the dialogues between Domain Experts and Knowledge Administrators progress, and allow greater detail and precision to be achieved. The goal is to be able to reach the Expression appropriate to the Rules format; Procedures or Formulas, such as in their Schemes of Records have already been seen.

As the Rules, Procedures or Formulas are developed that express the way to obtain the values that are required to complete each one of the 'Actions' registered, and the new Attributes considered in them are incorporated into the 'Repository' in one each records, the 'States' will go to '1' and, checking their



feasibility, those 'States' will go to '2'.

Figure 12 - Repository Records for Procedures and Formulas

When all of them are at '2', then the 'Action' part of Rule Y, whose status at the beginning of this analysis was equal to '0', will also go to their State ='2'. **[2]**

Thus, with **[1]** and **[2]** satisfied, the analysis of a Rule that ensures its operability in Production time culminates. Then, the Rule Status equals 3, as indicated in Figure 5.

Note 1: *What must be monitored very carefully, during the analysis, is that a situation such as the one indicated as 'Unsolvable' does not occur; Status = 6. This situation occurs when in the Domain of a Rule X, which precedes a Rule Z in its execution, an Attribute appears whose Value belongs to the image of Rule Z, known as' deadlock 'or' mortal embrace '. It means that the path chosen to obtain the necessary Values is not the correct one.*

Note 2: *Values that were associated with "System 'have been shown. Just as the 'Parameter' Attribute* *is associated with the Entity 'Domain' of the Rules, and it includes all those that operate with its Values. Just as 'Procedure (Id)' is associated with the Entity 'Procedure', the Attributes associated with 'System' will be those that operate depending on their operating environment. An example of this is 'Date (today)' = 'Today's date'; 'Date of the day'. These Entities can be represented as such in the Repository, if from any component of them, Attribute Values considered in some of the components included in the "Domain" are derived.*

5.2. EXPLANATION FACILITIES.

Since an instance of 'Procedure' is started, everything that has happened is registered in its records in the same order in which the sequence of its process flows, as indicated in each Rule, Procedure or Formula, as well as the successive partial results, until reaching the result that ends that instance of Procedure. In the case of the example, if it were a new 'Benefit', a message would be triggered towards the Entity that models its behavior, the way to start its 'life cycle'.

All the records related to this 'Procedure' are part of the 'Explanation' of the Final result and can be kept as a document of certified validity, when the administrative instances that the management indicates as necessary to give it institutional validity are perfected. Those Instances may also be incorporated into the corresponding Rule (BR). As mentioned before, the Digital Signature can be a valuable auxiliary that integrates everything mentioned in this item.

6. CONCLUSIONS

As a Conclusion of everything exposed up to here, a Proposal and two selected texts are offered for reflection.

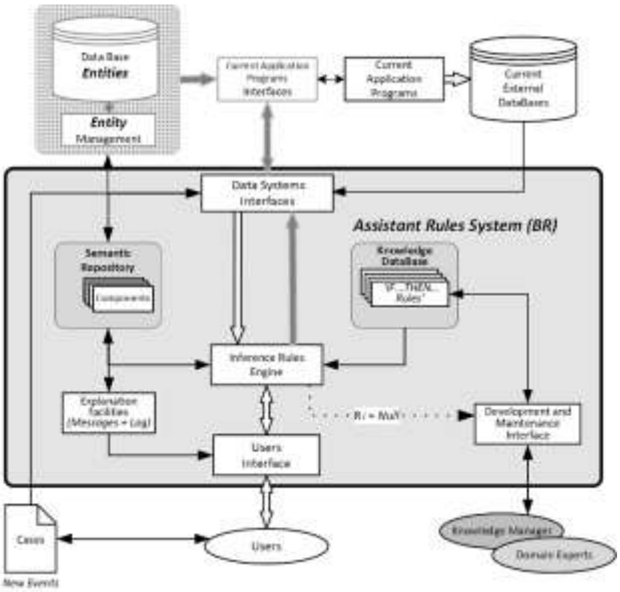
6.1. PROPOSAL

It is an architecture where the LATC algorithm is inserted into a system that: integrates the Knowledge Acquisition, through the records of a Repository, makes it possible to obtain a set of complete and consistent Rules (BR), offers the opportunity to test the obtained Rules, incorporating the Results or Actions that are pertinent to the added instances, and a substitute for Rules or Inferences Engine (more efficient than the one mentioned in the literature in this regard), which can operate in facing the demand for services from Clients, users, or Citizens, together with individual user interfaces, and for the Group of Experts, as well as interfaces to and from the Data Systems, which provide some of the data registered in their Bases and that they are required for the operation of the Rules. Interfaces that, in turn, can receive the results of the logic of decisions executed through these Rules, and execute the computer processes that correspond to those decisions.

If this architecture is implemented, it can be considered that what is proposed in this work constitutes an effective way to obtain the purposes initially set forth, with an indispensable minimum of integrated and shared effort, aimed at a common Organizational goal.

The relatively simple architecture depicted in the diagram in Figure 14 shows how it would be possible for the logic of management decisions and their procedures to become part of the domain of those who Manage the affairs of an Organization, which would give full satisfaction to what the BRG proposed in its Manifesto [1], for that resource.

The proposal is technically feasible and its implementation can be incremental and evolutionary. Its development is possible with the computer resources with which, almost certainly, the Organization already has.



It can be seen as a start in the re-engineering of data processes or, also, of business processes.

The Repository, the LATC algorithm and the way of operation of the Rules are its keys to it.

Figure 13 - Complete architecture for an integrated Rules (BR) Based Expert System).

It does not involve a new technological paradigm, but it does have an organizational aspect. [Khun, T. S. (11)].

What will surely be required, as an essential matter, is that there be in the Organization the indispensable number of people with their minds open, to successfully face the task of its development and implementation.

In this regard, let us introduce the following two texts for reflection.

6.2. TEXTS TO REFLECT ON WHAT IS DISCUSSED IN THIS WORK.

"THE ORGANIZATIONAL LEARNING OF ARGYRIS AND SCHÖN".

In the conceptual scheme of Argyris and Schön, simple circuit learning occurs when members of an organization cooperatively respond to changes in the internal and external context of the organization, detecting errors that they can correct, evaluating and generalizing the results while maintaining the central features of the theory to use. This type of learning allows the organization's rules to be maintained unchanged. The second type, dual loop learning, is capable of questioning itself as a learning system, it is a process of inquiry about the detection and correction of errors based on the inconsistency between the declared theories and the theories in use. This is a dynamic and changing process, aimed at transforming the organization's status quo. Argyris and Schön argue that there are two types of theories in organizations.

One of them is the so-called declared or official theory and the other theory in use. The first is constituted by the explicit rules of the organization that usually materialize in regulations and organization charts. While the other is the one that can be deduced from observation of what is actually done, that is, from action. There are usually inconsistencies between the two types.

Reducing this distance, seeking coherence between the two, is the task of organizational learning. "Organizational learning takes place when individuals within an organization experience a problem situation and investigate within it on behalf of the organization" ... "In order to become organizational, the learning that results from inquiry must be incorporated into the images of the organization retained in the minds of its members and / or in the epistemological elements (maps, memory, programs) in the organizational environment" ... [Argyris, C. & Schön, D (12)].

“COLLECTIVE MIND IN ORGANIZATIONS”.

“When individuals in the organization undergo an organizational research process, they travel a path where they articulate thought with action. This path, made possible by research, is the process of organizational learning”.

[Weick, Karl E. & Roberts, Karlene H (13)]

---o0o---

ISSN 2229-5518.

APPENDIX A

LATC ALGORITHM (LEARNING ASSISTANT TROUGH CARDINALITY)

Cardinality, Completeness and Consistency of the Rules(BR)

Let *'T'* be a set of Example Rules, consistent and free of redundancies.

Rule Id	Rule Name							
Instance	Conditions (Conditional Part)						Actions Part	Explanations
	At_1	At_2	---	At_j	...	At_n	Result or Derivation	
1	$a_{(1),1}$	$a_{(2),1}$	---	$a_{(j),1}$	...	$a_{(n),1}$	R_1	α (t)$_1$
2	$a_{(1),c}$	$a_{(2),4}$	---	$a_{(j),2}$	...	$a_{(n),2}$	R_2	α (t)$_2$
----	....	....	---	....	...	....	....	....
k	$a_{(1),b}$	$a_{(2),d}$	---	$a_{(j),p}$	...	$a_{(n),1}$	R_t	α (t)$_k$
----	....	....	---	....	...	....	....	....
m	$a_{(1),3}$	$a_{(2),z}$	---	$a_{(j),1}$	---	$a_{(n),q}$	R_s	α (t)$_m$

Figure 1 - Basic Outline of a Business Rule (BR)

Notation:

- With $'C_i^p'$, we will refer to the contents of a Table T or a Sub Table, where p will indicate its content, after an eventual instance p of selection($0 \leq p$); and where i, the Attribute index considered in that Table.
- $'C_i^0'$ It will then refer to an analysis of the complete Initial Table.
- $'Cp_i^p'$ It will denote the Complementary Table of T, such that $\{C_i^p \ \cup \ Cp_i^p\}$ satisfies the *Completeness Condition*

The following elements are defined:

- $A = \{a_1; \ a_2;;;; \ a_n\}$ the set of 'n' Attributes of a Table of Rules 'T'.
- $V(a_i) = \{(a_i)\}$ the set of values that an Attribute A_i can assume on each instance of Table C_i^0 of 'T'. *1<i<n; 1<j<m.*
- $C_j$each one of the *'j'* instances of Table *'T'*. *1<j<m.*
- $q \ (a_i)$ Frequency of value (a_i); $q \ (a_i)$ = $Freq_{(i)} = \sum_{(a_i)_j} 1, \ if \ ((z_i)_j \ \varepsilon \ Cj) \ and \ (z_i)_j = V_{((a)_i)_j}$.
- q (T), quantity of instances (*cases*) contemplated in a Table *'T'*.
- $\{R_i\}$ is the set of results that classify each C_j in one, and only one category R_i in *'T'*.
- We will note the Cardinality of $V(a_i)_j$ as $Card(A_i)$[6]. $\forall \ i, Card(A_i) \in N$.
- $Card$ [Ax, C_i^j] will denote the Cardinality of Attribute A_x in Sub Table C_i^j
- $Card[(A_i)]_p$: $Card(A_i)$ in the instance *p* of the selection. *p=0* will note the initial instance.

Taking on the definitions formulated by [Michalski (7)]:

If *'T'* satisfies $\forall \ i \ \in I \ (E_i \ \Rightarrow \ D_i)$ from [6]

and

$\forall \ i,j \ \in I \left(D_i \ \Rightarrow \sim E_j\right), if \ j \ \neq i$ from [7]

Where E_I ; *(i ∈ I),* is a description satisfied by all the example cases of the class K_I, and only for such examples. *(The logical disjunction of the different example cases)*

Therefore:

Because of [6], *'T'* satisfies the Completeness condition, and because of [7], the Consistency condition. [Michalski (7)].

FREQUENCIES TABLE FOR C_i^0

ATTRIBUTE	$Value_1$	$Freq_1$	$Value_2$	$Freq_2$	$Value_3$	$Freq_3$	V_i	F_i	$Value_n$	$Freq_n$	$Card(A_i)$
A_1	$(V_{a_1})_1$	$\sum 1$	$(V_{a_1})_2$	$\sum 1$	----	----	---	...	----	----	$Card(V_{a_1})$
a	$(V_{a_2})_1$	$\sum 1$	$(V_{a_2})_2$	$\sum 1$	----	----	...	...	----	----	$Card(V_{a_2})$
...	...	...	...	...	...	...	---	...	...	---	...
A_m	$(V_{a_m})_1$	$\sum 1$	$(V_{a_m})_2$	$\sum 1$	$(V_{a_m})_3$	$\sum 1$	---	...	$(V_{a_m})_n$	$\sum 1$	$Card(V_{a_m})$

Table 1 – Frequencies Table pattern – (Conditional part)

Note: $\boldsymbol{Freq_i}$= $\sum_{(a_i)_j} 1$, if $((z_i)_j \ \varepsilon \ C_j)$ and $((z_i)_j = V(a_i)_j$.

RESULTS TABLE FOR C_i^0.

It is analogously calculated, considering all the cases or instances of the Table and their Results. R_i.

	R_i				
Cases	V(r1)	V(r2)	...	V(rs)	$Card \ (R_i)$
$q \ (T)$	$\sum 1$	$\sum 1$	...	$\sum 1$	Card $(V_{(r)})$

Table 2 – Frequencies Table pattern – (Results part)

LEMMA 1

Let T be a Table or Sub Table Complete and consistent.

Let A_i be an Attribute of T; $V(a_i) = \{(a_i)\}$,

Therefore: $\forall A_{i,} Card \ A_i \ \geq 2$

PROOF.

Let's suppose $Card \ A_i \ < 2 \ \Rightarrow \ Card \ A_i \ = 1; \ if \ \{A_i\} \ \neq \emptyset$.

If $Card \ A_i \ = 1$, then $V(a_i) = \{(a_i)\}$ it has a single element. In other words: a_i single value.

Therefore

[6] *"We will call cardinal number to the general concept that, by means of our active ability to think, arises from the set M when we abstract from nature and the order of its elements. We denote the result of this double act of abstraction as Card (M)"*

Georg Cantor, 1895.

- Either T is incomplete. (*Contradiction*)
- Either T, the value for Ai turns out to be indifferent to the classification of the instances of T, or A_i *should not belong to* T (*Contradiction*).

LEMMA 2

If 'T' is complete and $a_i \in V(A_i) \Rightarrow \sim a_i \in V(A_i)$

It is the application of Lemma 1.

THEOREM 1

Thesis: If 'T' is complete $\Rightarrow \boldsymbol{q(T) = \prod_{i=1}^{n} Card(Ai)}$

Proof.

Let $\{(a_i)\}$ the set of 'n' elements of set A.

$V(a_i)_j = \cup_{j=1}^{n}\{(a_i)_j\} \Rightarrow Card\ V(a_i)_j) = \boldsymbol{Card}\ (A_i)$

Applying the combinatorial calculation, we have that the possible combinations of the elements of A, taken from 1 to 1, are represented by the expression:

$$C_1^{Card(A)} = \frac{[Card(A)]!}{1![[(Card\ (A)]-1])!} =$$

$$= \frac{[Card(A)]!}{[(Card\ (A)-1]!} = Card\ V(a) = Card\ A \qquad [a]$$

Let us now consider what Michalski [7] proposed in his statement of the Inductive Paradigm: "Find ...

…"A hypothesis H that tautologically implies the facts F, if F is a logical consequence of H, that is, if the expression H → F is true under all its interpretations.

('→' denotes a logical implication)."
[0]

Let H a hypothesis that satisfies the previous statement.

Let $[A_i]$ an expression of the form

$\{[A_i] = [(e_1)_i\ y\ (e_2)_i y \dots y (e_n)_i] \Rightarrow R_k\}$ where 'y' is the symbol of logical conjunction, and each (e_m) is an atomic expression that relates the name of an Attribute$[A_m]$, with any of the $(a_j) \in \cup_{i=1}^{n}\{(a)_j\}$. $(a_j) \in \{(a)_j\}$, which is the set of the values related to the Attribute $[A_m]$. $R_k \in \{(R)_j\}$, is the set of facts that symbolize the logical consequence of H, as each $[A_i], \forall\ i$.

Therefore $\forall h \in H, \exists [A_i]\ such\ that$

$(a_1)_i \in \{(a)_1\};\ (a_2)_i \in \{(a)_2\} \dots (a_n)_i \in \{(a)_n\};\ R_k \in \{(R)_k\}$
[b]

An isomorphism can be established between

$H\ and\ T \Rightarrow [A_i]_j\ isomorphism\ [T_j]$
[c]

Let T a Table that represents H; (T and H are isomorphic).

Let $T_j = [((a_1)_i)_j, ((a_2)_i)_j \dots ((a_n)_i)_j\ |R_k]$;

T_j *is the row 'j' of Table T*; T_j *is an 'instance' of T.*

Let A = {A_1, A_2,….A_n} the set of attributes of Rule Table T.

A_i *is the column 'i' of Tabla 'T'.*

A_1 *is the name of an Attribute of 'T'.*

Let $\{(a_i)\}$ the set of values than an Attribute A_i, can assume on each instance 'j' of Table T.

$V(a_i)_j = \cup_{j=1}^{n}\{(a_i)_j\};\ (1 < I < n)$.

By [a], $Card\ V(a) = Card\ A$.

[d]

Then, the totality of the possible interpretations of T required in [0], will be given by all the possible combinations of the different values of each one of the Attributes contemplated in T. Thus, considering what is expressed in [a] in [0] and in [d], that totality of possible interpretations of Hypothesis H, will be given by the expression:

$$C_1^{Card(A_1)} * C_1^{Card(A_2)} * \dots * C_1^{Card(A_n)} =$$

$$= Card(A_1) * Card(A_2) * \dots * Card(A_n) =$$

$$= q(T) = \prod_{i=1}^{n} Card_{(A_i)}$$

This expression represents the number of different instances that this Hypothesis must contemplate. This is so for the Conditional part of each instance of that Hypothesis. The tautology, in each case, must be supervised and verified by the Group of Experts, in accordance with mentioned by Michalski [7] in the preconditions of the Inference Paradigm:

"knowledge about the problem or background knowledge that defines the restrictions and affinities imposed on the given observations and allows the generation of candidate inductive hypotheses and that, in addition, may include some information on the theory of the domain that allows defining criteria preferably to characterize the desirable properties of the induced hypothesis".

So if 'T' represents a Hypothesis that satisfies the condition of **Completeness**, and being that if

$t_i, t_j \in T \Rightarrow t_i \neq t_j$, it should be that: If

'T' is complete $\Rightarrow q(T) = \prod_{i=1}^{n} Card(A_i)$. (■)Q.E.D.
[8]

COROLLARY 1

If $q(T) < \prod_{i=1}^{n} Card \Rightarrow$ 'T' is incomplete.
[9]

Proof

Applying "*modus tollendo tollens*" to the result of Theorem 1, we will have to:

$$q(T) \neq \prod_{i=1}^{n} Card\,(A_i).$$

Thus, two alternatives are proposed:

1. $if\ q(T) > \prod_{i=1}^{n} Card\,(A_i)$, then it will be an alternative to be dismissed, since it contradicts the conditions of consistency and non-redundancy considered as satisfied conditions.
2. if $q(T) < \prod_{i=1}^{n} Card\,(A_i)$ ⇒ No (T is complete) ≡ T is incomplete.

LEMMA 3

If for a Sub Table 'T', it is verified that q (T) = 1,then the classification of the only instance included in C_i^{p+1} and the analysis of the cases included in that classification branch, will have concluded.

PROOF

It is immediately concluded in the veracity of Lemma 3, since in such a case, T contains only one instance. Then there will be a single Category 'R_i' ε $\{Ri\}$ to which the only instance included in Sub Table T can be assigned.

THEOREM 2

Definition: Let $\{SA\}$ the set of Attributes selected in the 'p' previous iterations of Selection, so that: if A_i was selected ⇒ $SA_i \in \{SA\}$ y Card $(A_i) = Card[(SA_i)]_p$.

HYPOTHESIS

Let 'T' a set of Rules both consistent and free of redundancies.

THESIS:

Le T^p the result of the iteration of p^{th} selection on Table T. Let A_f the selected attribute, accumulating 's' selected Attributes, so that $Card\ (SA) =$'s'.

Therefore, if T satisfies the Completeness Condition, T^p will still be Complete if it verifies that:

$$if\ q(T^p) > 1\ (by\ Lema\ 3) \Rightarrow$$

$$\Rightarrow q(T^p) = \frac{\prod_{i=1}^{n}[Card(A_i)]_0}{\prod_{i=1}^{s}\,[Card(SA_i)]_p}$$

PROOF. (BY INDUCTION OVER 'P')

1. Let $p = 1$.

$q(T) = \prod_{i=1}^{n} Card(A_i) = Card(A_1) * \dots * Card(A_f) * \dots * Card(A_n) = \prod_{i=1}^{n} Card(Ai)$

Let A_f the selected Attribute.

$q(T') = \prod_{i=1}^{n} Card(A_i) = Card(A_1) * \dots * Card(A_n) =$

$$= \frac{\prod_{i=1}^{n}[Card(A_i)]_0}{Card(A_f)} = \frac{\prod_{i=1}^{n}[Card(A_i)]_0}{\prod_{i=1}^{s}\,[Card(SA_i)]_1}$$

2. $If\ V(p) \Rightarrow V(p+1)$

$Let\ p \Rightarrow p + 1 < n.$

$q(T) = \prod_{i=1}^{n} Card(A_i) = Card(A_1) * \dots Card(A_g) \dots * Card(A_n) =$

$$= \prod_{i=1}^{n} Card(Ai)$$

By inductive hypothesis, it will:

$$q(T^p) = \frac{\prod_{i=1}^{n}[Card(A_i)]_0}{\prod_{i=1}^{s}\,[Card(SA_i)]_p}$$

Let A_g the selected Attribute in $p + 1^{th}$ iteration. Therefore:

$$q\left(T^{(p+1)}\right) = \frac{\prod_{i=1}^{n}[Card(A_i)]_0}{\prod_{i=1}^{s}\,[Card(SA_i)]_p\, Card\,(A_g)} =$$

$$\frac{\prod_{i=1}^{n}[Card(A_i)]_0}{\prod_{i=1}^{s}\,[Card(SA_i)]_{p+1}}$$

From **1)** and **2)**: If T satisfies the *Completeness Condition*, T^p will continue to be Complete if it verifies that:

$$if\ q(T^p) > 1 \xrightarrow{Lemma\ 3}$$

$$\xrightarrow{Lemma\ 3} q(T^p) = \frac{\prod_{i=1}^{n}[Card(A_i)]_0}{\prod_{i=1}^{s}\,[Card(SA_i)]_p} \qquad (\blacksquare)$$

Q.E.D.

COROLLARY 2: *Indication to build* Cp_i^p

a) If, once we finished an iteration 'j' of the selection and $\{SA_i\} \neq \emptyset$, we will get:

If $\boldsymbol{q}\,C_i^j > 1$(*Lemma 3*) and $\boldsymbol{q}\,C_i^j < \frac{\prod_{i=1}^{n}[Card(A_i)]_0}{\prod_{i=1}^{s} Card(SA_i)}$ ⇒

⇒ ∃ $A_x \in (C_i^j)$ so that *Card* $[A_x, C_i^j] <$ *Card*$[A_x, C_i^0]$;

[10a]

Its practical meaning is: If at the end of iteration 'J' of the selection, it is verified that

$$qC_i^j > 1\ and\ qC_i^j < \frac{\prod_{i=1}^{n}[Card(A_i)]_0}{\prod_{i=1}^{s} Card(SA_i)},$$

then there will be at least one Attribute $A_x \in (C_i^j)$ *with Card* (A_x) *lower than the one obtained for* C_i^0.

Then, there will be a complementary Table of Rules for C_i^p, denominated Cp_i^p so that

$\{C_i^p \cup Cp_i^p\}$ satisfies the ***Completeness Condition***.

b) If$\{SA_i\} = \emptyset$ (*As would be the case when applied directly to the Initial Table*C_i^0, therefore:

if $\boldsymbol{q}\,C_i^j > 1$(*Lemma 3*) and $\boldsymbol{q}\,C_i^j < \prod_{i=1}^{n}[Card(A_i)]_0$ ⇒

⇒ ∃ $A_x \in (C_i^j)$ so that

Card $[A_x, C_i^j] <$ *Card* $[A_x, C_i^0]$;

[10b]

Whose practical meaning is: If when starting the Cardinality Analysis there was no Attribute selection and $\{SA_i\} = \emptyset$, then:

We will verify that if: $qC_i^j > 1$ and

$qC_i^j < \prod_{i=1}^{n}[Card(A_i)]_0$, then there will be

$A_x \, \varepsilon \, (C_i^j)$ so that the value for Card (A_x) in sub Table (C_i^j), is lower than the one obtained for C_i^0.

As such, the instances of a supplementary Table of Rules must be proposed. C_i^p : Cp_i^p so that $\{C_i^p \cup Cp_i^p\}$ satisfies the **Completeness Condition**.

COROLLARY 3: Quantity (size) of Cp_i^p.

a) if $\{SA_i\} \neq \emptyset$, it has to:

if $qC_i^p > 1$ and $qC_i^p < \frac{\prod_{i=1}^{n}[Card(A_i)]_0}{\prod_{i=1}^{s} Card(SA_i)} \Rightarrow$

$\Rightarrow \exists \; Ax \in C_i^p \; tq \; Card[Ax, C_i^p] < Card[Ax, C_0^0]$

Let kC_i^p tq $[qC_i^p + kC_i^p] = \frac{\prod_{i=1}^{n}[Card(A_i)]_0}{\prod_{i=1}^{s} Card(SA_i)} \Rightarrow$

$\Rightarrow kC_i^p = \frac{\prod_{i=1}^{n}[Card(A_i)]_0}{\prod_{i=1}^{s} Card(SA_i)} - qC_i^p$.

[11a

]

Therefore kC_i^p is the quantity of instances ***that must be added*** to achieve Completeness of C_i^p .

If $\{SA_i\} = \emptyset$ (As would be the case when applied directly to the Initial TableC_i^0, then, applying [10b]:

If $\boldsymbol{q}\,C_i^j$>1 and $\boldsymbol{q}\,C_i^j < \prod_{i=1}^{n}[Card(A_i)]_0 \Rightarrow$

$\Rightarrow \exists \; Ax \; \varepsilon \; (C_i^j)$so that *Card [Ax, C_i^j]<Card[Ax, C_i^0];*

Let kC_i^p tq $[qC_i^p + kC_i^p] = \prod_{i=1}^{n}[Card(A_i)]_0 \Rightarrow$

$\Rightarrow kC_i^p \;=\; \prod_{i=1}^{n}[Card(A_i)]_0 - qC_i^p$.

[11b

]

COROLLARY 4: Contents of Cp_i^p.

Let $\{SA_i\} \neq \emptyset$, then:

Let Ax the Attribute that verifies Corollary 1a) [10a] and $\{(ax)_1, (ax)_2 ... (ax)_t\}$ the set of Values for Ax; Card $q(Ax) = t$.

Let $k(Ax)$ the quantity of instances that should be generated for Cp_i^p y $k(Ax)_i$, the quantity of those whose Attribute Ax contains the value $(a_x)_i$;

$;(a_x)_i \in \{(a_i)\}$.

Contents of Cp_i^p	Attribute in 'deficit'	Missing Instances	Attribute Values to consider	
Cardinality of C_i^p in completeness	Ax	$\sum_{y=1}^{t} k(ax)_y$	$(ax)_1$	$k(ax)_1$
$Card\ C_i^p = \frac{\prod_{i=1}^{n}[Card(A_i)]_0}{\prod_{i=1}^{p}[Card(SA_i)]_p}$ *			(ax)2	k(ax)2
			—	—
• *If $Card(SA_i) \geq 1$); if not: **[11b]**			$(ax)_t$	$k(ax)_t$

Table 3 – Deficit Table of Contents pattern

Therefore,

$\forall y, \; k(Ax)_y = \frac{Card\ (C_i^p)}{Card\ [(Ax), C_i^0]} - Frec\ [(ax)_y, C_i^p]$

[12]

It's easy to note that, for every$(ax)_i$, we are obtaining the complementary number of instances and frequencies that should consider that value of the Ax Attribute, so that $\{\{C_i^p \cup Cp_i^p\}\}_{Ax}$ satisfies the **Completeness Condition**, as far as that Attribute is concerned.

That is, by Theorem 1, [8]:

If 'T' is complete $\Rightarrow \; q(T) \; = \prod_{i=1}^{n} Card\ (Ai)$

The Table of Contents is completed, adding to each of the rest of the Attributes in succession, one after the other, after having exhausted the possible values for the previous one.

Now, let 'A_z' and A_w the Attributes that have not been analyzed until now.

(They should be added in the 'Order' in which they appear in the Frequency Table C_i^0).

Contents of Cp_i^p	Attribute in 'deficit'	Missing Instances	Attribute Values to consider	
Cardinality of C_i^p in completeness	A_x	$\sum_{y=1}^{t} k(ax)_y$	$(ax)_1$	$k(ax)_1$
$Card\ C_i^p = \frac{\prod_{i=1}^{n}[Card(A_i)]_0}{\prod_{i=1}^{p}[Card(SA_i)]_p}$			$(ax)_2$	$k(ax)_r$
			—	—
•*If $Card(SA_i) \geq 1$); if not: **[11b]**			$(ax)_t$	$k(ax)_t$
	Attributes not selected			
(*) *The value of the missing instances will not be obtained again, since it would be the same value for any of the Attributes.*	Az	(*)	$(az)_1$	$k(az)_1$
			—	—
	Aw	(*)	$(aw)_1$	$k(aw)_1$
			—	—
			$(aw)_u$	$k(aw)_u$

Table 4 - Table of Contents in 'deficit'

ASSEMBLING THE TABLE Cp_i^p – LATC ALGORITHM

In order to proceed with assembling the table, the steps of the algorithm can be followed in this manner:

1. To avoid the formation of redundancies or inconsistencies, for the assembly of the instances of the Table C_i^p we will obtain the Table of Frequencies that would correspond to $\{C_i^p \cup Cp_i^p\}$.

Given that $[C_i^p \cap Cp_i^p] = \emptyset$, Frequency $[C_i^p \cup Cp_i^p] = Freq\ C_i^p + Freq\ Cp_i^p$									
ATTRIBUTE	Value$_1$	Freq$_1$	Value$_2$	Freq$_2$	Value$_3$	Freq$_3$	Value$_n$	Freq$_n$	Card.
A_1	$V(a_1)_1$	$q(a_1)_1 + k(A_1)_1$	$V(a_1)_2$	$q(a_1)_2 + k(A_1)_2$	—	—	—	--	2
A_2	$V(a_2)_1$	$q(a_2)_1 + k(A_2)_1$	$V(a_2)_2$	$q(a_2)_1 + k(A_2)_2$	--	—	_	--	2
...	—	—	—	—	—	—	—	..	—
A_m	$V(a_m)_1$	$q(a_m)_1 + k(A_m)_1$	$V(a_m)_2$	$q(a_m)_2 + k(A_m)_2$	--	–	$V(a_m)_n$	$q(a_m)_n + k(A_m)_n$	n

2. As follows:

Table 5 - Assembled Table $\boldsymbol{Cp_i^p}$ – LATC Algorithm

1. For the '*Conditional*' part, the Attributes considered in the Frequency Table C_i^0 are taken into account and in the same order in which they appear in it. Each Attribute in a Column, as they appear in Table C_i^p.
2. As many instances (rows) of the Table are generated as indicated in the '**Missing Instances**' column.

a. For the 'Conditions' part, the Attributes considered in C_i^0 are taken into account and in the same order in which they appear in it (The same of the rows of the Table 'Added Frequencies'). Each Attribute in a Column, as they appear in the Table C_i^p. Let 'm' be the number of Attributes considered in Table C_i^0.

b. As many instances (rows) of the Table are generated as indicated $\prod_{i=1}^{n}[Card(A_i)]_0$. *Consequently, the Table will contain 'm' columns for the Attribute values and a last column, reserved for the 'Actions' or 'Results' part of the Table.*

c. To effectively complete the values relevant to the cells corresponding to the columns 'Attributes' columns that make up the 'Conditions' of the new Table, you can proceed as follows:

1. Initial value (J=1); ('m' = Quantity of significant Attributes).
2. Initial value (i=0); ($SCR_j = 0$)
3. Calculate $CR_j' = \frac{Card\,C_i^0}{\prod_1^j Card(A_j)}$; ($CR_j$: Quantity of repetitions of the value of an $Attribute_j$; *1≤j≤m*)
4.

   i. If 'i' $= Card\ (Aj) \rightarrow (i = 1)$ If not, then $(i = i + 1)$

   ii. Assign value 'CR_j' instances of Column 'j' with '$V(a_i)_j$'

   iii. SCR_j = 'SCR_j' + 'CR_j'

   iv. If 'SCR_j' $< \prod_{i=1}^{n}[Card(A_i)]_0$, iterates from c.4.i)

4. If $SCR_j = \prod_{i=1}^{n}[Card(A_i)]_0 \Rightarrow$ (j=j+1).
5. If $j \leq m$, iterates from c.2). If $J > m \overset{end}{\Longrightarrow}$ <u>the Table has been completed</u> in its '***Conditional***' part.

6. Finally, the column corresponding to 'Results' or 'Actions' will be added, having as the content of their cells, the symbol deemed appropriate, for detection by the Group of Experts.
7. In each of the corresponding instances, in its 'Conditional' part, with the one that is part of the Initial Table, that symbol will be replaced by the pertinent value specified in that Initial Table. [7]
8. The rest of the instances of the New Table will keep that symbol, indicating the ignorance of the value of each of those results.

At some point, the Group of Experts must incorporate the value of the "Result", to each of the proposed instances, as results of the application of the LATC Algorithm. and added to the standard Initial Table of Examples, in accordance with the interpretation it makes of the values of the 'Conditions'.

After that, the consistency analysis of the new Table formed can be addressed.

CONSISTENCY

The final test of the Consistency Condition can only be carried out after the values of the results corresponding to each of the proposed instances have been assigned, added to satisfy the Completion Condition. This assignment will be made according to the reference mentioned in Theorem 1 that, on Background Knowledge, is mentioned by [Michalski, 7].

ANOMALIES IN THE RULE DOMAIN.

A possible change in the Attribute Values will imply the need to rerun the LATC Algorithm with the new set of Values. We have already mentioned the possibility of its occurrence at operating time.

What is feasible to do is to proceed in such a way that the Group of Experts can systematically become aware of the possible incidence of a problem of this nature.

To do this, it would suffice to add to the possible values for each Attribute, one referring to "Out of Range", which would be equivalent to 0 (zero) in the last cell of each column of Table Attributes. As 'Result' for that instance of the Table, a referral to the Group of Experts would be determined and in the corresponding text, a message in which the operator would add the name of the Attribute (s) involved and the value that fell outside the range allowed for such an Attribute.

The Expert Group will check whether it is a mistake or an anomaly. In the latter case, it would proceed to the correction of the Table, in accordance with the internal norm that regulates that type of situation.

---oo0oo--

[7] *This is the condition for which it is considered convenient to maintain the same order in all the Generated Tables, both of Contents and Frequencies. It is necessary to maintain the possibility of matching the Conditional part of the Table, in order to replicate the Values of the 'Results' without risk of altering the meaning of the Example provided by the Experts. Alternative: match as 'Case' with Instance.*

BIBLIOGRAPHY

[1] BUSINESS RULES GROUP, © 2003 – [BUSINESS RULES MANIFESTO – (*The Principles of Rules Independence*)] – Version 2.0, Nov. 2003 – Edited by Ronald G. Ross - available at http://www.businessrulesgroup.org/brmanifesto.htm

[2] Young, Richard M - Production Systems in Cognitive Psychology - In the 'International Encyclopedia of the Social & Behavioral Sciences', 2001 – available at http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.12.9158&rep=rep1&type=pdf

[3] Xindong Wu – "Inductive Learning: Algorithms and Frontiers" – Artificial intelligence Review 7, 93--108, 1993. © 1993 Kluwer Academic Publishers. Printed in the Netherlands. Available at: https://doi.org/10.1007/BF00849079'

[4] Feigenbaum, E. – "The Handbook of Artificial Intelligence", Vol. 1, edited by Avron Barr, Edward A. Feigenbaum, Jan. 1981.

[5] Jones, P.H. 1989. "Knowledge Acquisition" In: Barrett, J.R. and D.D. Jones. Knowledge Engineering in Agriculture. ASAE Monograph No. 8, ASAE, St. Joseph, MI. – Available at: https://engineering.purdue.edu/~engelb/abe565/knowacq.htm

[6] Negnevitsky, M. – "Artificial Intelligence - (A Guide to Intelligent Systems)" – 2nd Edition (2005) – Addison Wesley - Pearson Education Limited, Harlow, Essex – England.

[7] Michalski, Ryszard S. – "Readings in Machine Learning". Originally posted on "Artificial Intelligence" 20 (1983) pp.111-116 – available at: http://ebot.gmu.edu/bitstream/handle/1920/1654/90-43.pdf?sequence=1&isAllowed=y

[8] Allen Newell & Herbert A. Simon (1976) – "Computer Science as Empirical Inquiry: Symbols and Search" in Communication to ACM - March 1976; Volume 19; Number 3. Recovered from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.104.2482&rep=rep1&type=pdf

[9] Johnson, Jeff – "Designing with the Mind in Mind" - Morgan Kaufmann Publishers is an imprint of Elsevier - © 2010 Elsevier Inc.

[10] Raggio, Raul A. – "Objetos de Gestión" - © 2004/6 Editorial Raggio - Buenos Aires. Available at: https://books.google.com.ar/books/about/Objetos_de_Gestion.html?id=4GD_yxmBHtQC&redir_esc=y
A new version is available; "Objetos de Gestión II", which contemplates the content of this document and places it within the context of the Management Object model: the 'OG Model'. Available at:
https://raulangelraggio.academia.edu/research

[11] Khun, T. S. – La estructura de las revoluciones científicas – ©Fondo de Cultura Económica (1971-1988) - ©1962 University of Chicago Press.

[12] Argyris, C. & Schön, D.: "Organizational Learning: A theory of action perspective". In 'Revista Española de Investigaciones Sociológicas: Reis'. 77-78/97 pp. 345-348. Available at: http://213.134.36.200/REIS/PDF/REIS_077_078_19.pdf

[13] Weick, Karl E. & Roberts, Karlene H. - Collective Mind in Organizations: Heedful Interrelating on Flight Decks. © 1993 by Cornell University. Available at: https://www.semanticscholar.org/paper/Collective-mind-in-organizations%3A-Heedful-on-flight-Weick-Roberts/4ce50aca5cb5b3a7d6715af171cc6b4a4d46d720

[14] Pareto, V. – Juran, J.M. - The Non-Pareto Principle; Mea Culpa. Available at: https://www.semanticscholar.org/paper/The-Non-Pareto-Principle%3B-Mea-Culpa-Juran/587b2c46da9fd6f63d50d992bdb70dabbad39407.