Using Paraconsistent Logic in a Multi-Agent System

Elaini S. Angelotti*, Edson E. Scalabrin, Bráulio C. Ávila
Pontifical Catholic University of Paraná, Brazil
Department of Informatics

Abstract *This work is part of the Multicheck Project that defines architecture of autonomous agents for the automatic treatment of handwritten Brazilian bank checks. The competence of these agents is implemented in two layers. The first one corresponds to algorithms of patterns recognition algorithms directly applied on the image segment. The second one corresponds to reasoning mechanisms applied to the information from the first layer either to validate or to interpret it. The interpretation process involves as well information obtained from other agents. Therefore, information can present inconsistencies. This problem is treated properly and naturally through concepts and operators of paraconsistent logic. This paper focuses on the second layer, on task distribution problems and on communication between agents. The first layer information was obtained through a simulated database.*

*Keywords***:** *Artificial Intelligence, Autonomous Agent, Paraconsistent Logic, and Task Distribution*

1. Introduction

In a bank environment, the manual verification of checks by employees, in spite of being a trivial task, can cause some problems such as: technical incapability, person in charge's ability, delay in accomplishing tasks, etc. The automation allows a faster and more reliable processing of the task, offering reduction on costs as well as on compensation time. However, the automatic treatment of handwritten checks is a complex problem. Such a complexity as described by Scalabrin et al. [10], occurs because of the diversity and complexity of the involved knowledge, the need to reconfigure dynamically a treatment process, and the interaction between experts. The automation process requires the implementation of the operations follow:

(a) image acquisition;
(b) suppression of irrelevant information given on the check;
(c) relevant information location and extraction;
(d) obtaining of the document logical structure;
(e) discrimination between the pre-printed and the handwritten information;
(f) segmentation of each logical field;
(g) logical data interpretation (date, numerical, literal and signature);
(h) check analysis for acceptance or rejection.

Clearly, it is a problem which tasks are well defined. However, the implementation of each one requires large computer resources and the sharing of some partial results can be decisive on obtaining a correct interpretation of information.

Therefore, we decided to automate the bank check compensation process, using the concept of autonomous agent. This concept allows us to organize the application knowledge and brings several own benefits to the approach. Such approach was chosen for the following motivations:

(a) the nature of the problem in question allows a decomposition in well-defined tasks, and each of them can be encapsulated in an independent agent;
(b) the natural capability of interaction of the agents makes the check treatment process more robust, particularly as their exchanges solve situations which are apparently difficult;
(c) the possibility of introducing learning and reasoning mechanisms in the agents, allows us to endow them with pro-activated and adaptable behaviors;
(d) the modular aspect of the agents allows to fight effectively against the complexity of the domain, as well as it permits to develop a system in an incremental way, which means, an open system of agents [11].

Therefore, in a DAI (Distributed Artificial Intelligence) system, because of its distributed and non-synchronized nature, the agents can easily obtain inconsistent information working separately on the same problem. This way, some of these agents must be complex enough to decide *how*, *when* and *with whom* to interact and behave correctly facing contradictory

* *Supported by CAPES/Brazil*

information. The mechanism developed for this purpose uses some of the concepts and operators of paraconsistent logic, which integrate naturally inconsistent information treatment, that cannot be treated through a classic logic [1], [2], [4], and [12].

The section 2 presents architecture of autonomous agents that takes into account this interaction in a very natural way. The next sections describe the system operation, enhancing the mechanisms of combination and interpretation (or validation) of the information given by the image segments classifiers of a check logical field. It is important to remind that the communication and the validation process work together, allowing the agents to exchange beliefs and to reason about them. To finish, the work of Montoliu [9] and the conclusion of the ours work are discussed.

2. Architecture

The architecture of the *Multicheck System*[1] consists in a group of relatively complex agents turned to the analysis and treatment of handwritten Brazilian bank checks images [10]. In this architecture, four types of agents are defined: (i) t*he segmentation agent* identifies extracts and creates a logical model of a check. (ii) t*he recognition agent* recognizes the different logical fields extracted from a check (date, signature, numerical and literal value). (iii) t*he analysis agent* accepts or rejects a check. The task consists in verifying if all recognition agents have either or not given a positive interpretation of the same check. (iv) t*he manager agent* is responsible to monitor the net and decide if an agent should be inserted or removed from the system.

The Fig.1 shows a simple view of the Multicheck System Architecture, as well as the architecture of each of its agents. The ability to recognize patterns – over image segments – is present only in agents: *date, signature, numerical* and *literal*. The expertise to interpret and validate the patterns appear in all agents, except in the *segmentation agent*. The check acceptance or rejection is done by the *analysis agent*, which validates the information given by every *recognition agent*. The communication ability is present in all agents and is implemented by the communication module. This module is responsible for the exchange of non-synchronized messages between agents, and for the implementation of some basic tasks, such as: the recognition of a performative, the extraction of the message contents, etc.

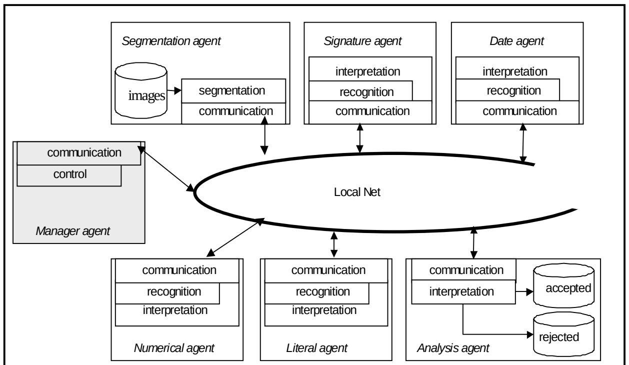


Fig. 1. Multicheck Architecture

It is important to remember that in the implementation of this architecture, there can be several agents implemented with the same competence. This redundancy allows us to aim for several parallel treatments and ensure the balance of the system load. However, the architecture has to have at least six agents (one of each type) to interpret a check.

In order to manage the balance of the system load was introduced a *manager agent* which is responsible to monitor the agents of the net. The main tasks of this agent consist in insert or remove agents from the system when necessary. This decision is take over the average time spent by one agent to end its calculus over a certain task. The ordered pair < *i*, *t* > correspond to information used by the manager to its take of decisions, where *i* is any agent and *t* is the average time spent by the agent to end its recognition task, as shown in Fig.1. For example:

$$< i, t > = \{< \text{signature}, 32s >, < \text{date}, 30s >, < \text{numeric}, 80s >, < \text{literal}, 90s >\}$$

[1] *The Multicheck Project is being developed by Pontifical Catholic University of Paraná, Brazil, with the financial support of the Brazilian Government (CNPq), in an international cooperation between l'École de technologie supérieure/Canadá and PUCPR*

The decision of insert or remove a recognition agent is take by the manager agent considering the value βi. The calculation of βi is obtained of following form:

(a) A = {32s, 30s, 80s, 90s}
(b) For each element of A do:
    $\beta_i = (A_i / \mathrm{Min}(A)) - 1$
    apply rule 01

The manager agent makes its decisions evaluating the following rules:

```
Rule 01: insert a new recognition agent in the system
    If <(βi > 0)> then <insert βi agents of the type Ai in the system>

Rule 02: insert a new analysis or segmentation agent in the system
    If <(numbers of checks in the queue > 50)> then <insert a new agent in the system>

Rule 03: remove an agent in the system
    If <(time of idleness of a agent > 1 minute)> then < remove the agent from the system>
```

The main advantage of the Multicheck Architecture resides on the autonomous and cognitive agents. These entities are able to communicate and reason about beliefs, turning the interpretation process of a check more robust, beyond allowing the repetition of treatment stages (if necessary). On the other hand, the biggest inconvenient consists in the complexity of the implementation of these agents, especially regarding the management and the treatment of its communication. For example: when and how an agent must communicate an information? When and how an agent must ask for an information? When and how the agents must organize themselves to accomplish the same goal?

3. Scene

The *numerical* and *literal agents* represent the most interesting aspect of this work, because the interpretation of the numerical and literal logical fields can be done in an interactive and approximate way, enabling these agents to exchange beliefs and reason about them. The Fig.2, shows summarily the working process of these agents.

Each *recognition process* corresponds to the range of classification algorithms applied on a certain logical field. The input of these processes are images and the output are pairs <*n*,[μ, ν]>, where μ represents the favorable evidence and ν the opposite evidence[2] on which *n* must be a digit in case of an *numerical agent* or a word in case of an *literal agent*. Each set of patterns obtained in a recognition process, is the input for an *interpretation process.*

The interpretation process of each pattern sets is realized in an interactive way, where, for example, the numerical and literal agents exchange information to solve certain internal conflicts and reach an agreement on the value of the check. These agents communicate their conclusions to the analysis agent, which accepts or rejects the conclusions (or interpretations). The decision is based only on favorable and opposite evidential values about information given by recognition agents. The result is obtained by the application of some operators of paraconsistent logic on these values, as well as by using some domain heuristics.

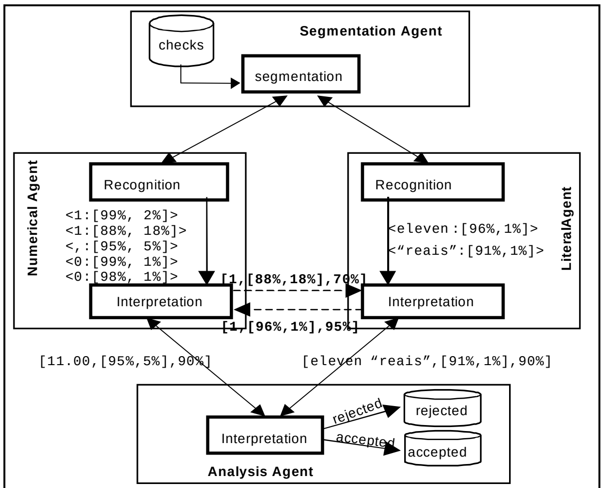


Fig. 2. Segmentation, recognition and validation of logic fields of numerical and literal values

It is important to remember that this work focuses on the validation of patterns obtained in recognition process, thereby it only concerns the implementation of the interpretation modules. The evidential values associated to the literal and numerical values were obtained using an automatic data generator. The various modules of recognition are part of the following works: signature [5], date [7], numerical value [3], literal value, and segmentation [8].

[2] *As Subrahmanian [16] says, the use of two evidences associated to a same* p *proposition, can reinforce its expressive capacity.*

3.1. Pattern interpretation or validation

The interpretation of check information is an interactive, approximated and distributed task, therefore it is not limited to a merely local process. Each agent implements this task supported by a high-level communication protocol. This protocol activates responding to the state of each agent and its local knowledge. This knowledge is encapsulated in the decision process of each agent.

During a check logical field process, concepts of evidential logic reasoning were used. In this type of reasoning, described by Subrahmanian [12], two values are associated to a proposition: one of them represents the favorable evidence to the proposition and the other one the opposite evidence [2] and [9]. No restriction is set to these values, except that they belong to interval [0,1]. In evidential logic favorable and opposite evidences factors aren't directly related as in the Probability Theory [5].

In summary, the logical field process of a check, follows a determined flow: the recognition module of a certain agent receives an image segment σ_i – which corresponds to a certain logical field of a check, and decomposes σ_i in various parts σ_{ij}. These parts are classified through highly specialized classifiers. Its output format is $< \sigma_{ij} \in N_k : [\mu_j ; \nu_j] >$, where $\mu_j, \nu_j \in [0, 1]$, and represents coefficients of favorable and opposite evidences in relation to the class that contains a determined σ_{ij}. N_k are the possible classes.

Given σ_1 the numerical value logical field, σ_{1j} the values of favorable and opposite evidence of each digit, and χ_{1j} the degrees of certainty, as shows Fig.3.

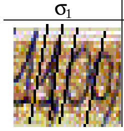
σ_1	σ_{1j}	χ_{1j}
	< 1 : [0.96 ; 0.01]>	0.95
	< 1 : [0.86 ; 0.01]>	0.85
	< , : [0.70 ; 0.25]>	0.45
	< 0 : [0.85 ; 0.36]>	0.49
	< 0 : [0.70 ; 0.30]>	0.40

Fig. 3. Image segment, degrees of favorable and contrary evidences, and certainty degrees.

For example, σ_{11} can be read as follows: there is a favorable evidence, up to 96%, that the first digit is "1", and an opposite evidence, up to 1%, that this first digit is not "1".

The evidential values interpretation is done through operators and paraconsistent logic concepts, where the evidences are mapped in certainty degrees through the following function [2], [9]:

$$c([\mu j, \nu j]) = \mu j - \nu j = \chi ij \qquad (1)$$

a certainty degree χ_{ij} associated to each classified σ_i segment. χ_{ij} shall be used in various situations, as to define when an agent must communicate with the others.

Example 3.1: valid rules for numeric and literal agents

```
Rule 04: If <χij ∈ (50, 90]> then <asks for
information to the literal or numeric agent to
increase χij>
Rule 05: If <min(χij) ∈ (90, 100]> then <sends the
result to the analysis agent and others
interested>
Rule 06: If <χij ∈ [0, 50]> then <asks for another
segmentation σi+1>
Rule 07: If <the request for a new segmentation
is rejected>
         then <concludes that the numerical value
cannot be recognized>
         and <sends the result to every other
agents>
[...]
```

Example 3.2: Rules for analysis agent

```
Rule 08: If   <one of the logical fields cannot
be interpreted correctly>
         then <rejects check> else <accepts
         check>
[...]
```

The thresholds presented on the rules above are suppositions. In particular, an agent searches an interaction when he cannot recognize the logical field of its competence, it can decide to: (i) *ask a segmentation agent* to take a new extraction of the logical field; (ii) *ask a recognition agent* to validate a belief; (iii) *warn all system agents* that the logical field of its competence couldn't be recognized.

The exchange of information between agents can result in new evidential coefficients, especially through successive combinations, which occur at two different moments: (i) during a local segmentation of a given logical field; (ii) during the interpretation of two or more logical fields that interact with each other.

Phase 1: combination of different segmentations and classifications on the same logical field

The *segmentation agent* identifies, extracts and creates the logical structure of a check (date, signature, literal and numerical value). In the first place the check global segmentation is realized, immediately followed by a local segmentation. This procedure allows any agent to ask the *segmentation agent* for a new extraction of a determined logical field. The recognition algorithms are applied to this new extraction, obtaining new evidential values and

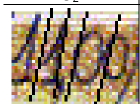
σ_2	σ_{2j}	χ_{2j}
	< 1 : [0.99 ; 0.02]>	0.97
	< 7 : [0.98 ; 0.01]>	0.97
	< , : [0.90 ; 0.11]>	0.79
	< 0 : [0.80 ; 0.23]>	0.57
	< 0 : [0.99 ; 0.40]>	0.59

Fig. 4. Second segmentation of the numerical amount, degrees of favorable and contrary evidences, and

certainty degrees, which are consequently combined.

On Fig.4, the third, fourth and fifth components of σ_1 were recognized with certainty degrees lower than 50%. Applying *Rule 06*, a new segmentation is requested.

Given σ_2 a new segmentation for the numerical value of the logical field, σ_{2j} the values of favorable and opposite evidence for each digit, and χ_{2j} the certainty degrees, as shows Fig.4.

$(\sigma_{1j,}\sigma_{2j})$	(χ_{3j})
< 1 : [0.99 ; 0.02]>	0.97
< 1 : [0.86 ; 0.01]>	0.85
< , : [0.90 ; 0.11]>	**0.79**
< 0 : [0.80 ; 0.23]>	**0.57**
< 0 : [0.99 ; 0.40]>	**0.59**

Fig. 5. Image segment, degrees of favorable and opposite evidences, and certainty degrees.

Each σ_{1j} value of the first segmentation (Fig.4) is compared to each σ_{2j} value of the second segmentation (Fig.5). If, for example, σ_{11} and σ_{21} belong to the same class, apply the supreme operator (*sup*) over χ_{11} e χ_{21}. The σ_{ij} that owns the highest certainty degree is selected. In this way, for σ_{11} and σ_{21} selects <1: [0.99 0.03], 96%>. The supreme operator is used because it returns the highest degree of certainty in the selective process. However, if σ_{11} and σ_{21} do not belong to the same class, it is necessary to begin the process of information exchange between *numerical and literal agents* to discover which classification is correct. It is important to remind that even if the certainty degree of σ_{22} is higher than the certainty degree of σ_{12}, σ_{12} will be selected. This occurs because the literal value is more decisive than the numerical[3] value. In this case, the combination of the results to σ_1 and σ_2 will be showed in Fig.6.

Phase 2: sharing of partial results from different logical fields

The sharing of partial results is fundamental between *literal* and *numerical agents*, especially because they must obtain exactly the same information from different logical fields. They can also obtain conflicting results and be leaded to interact with each other, to obtain a consistent interpretation and increase its certainty degree.

Assuming that the *literal* and *numerical agents* have already concluded independently *Phase 1* and have recognized the same information, so the consequent of *Rule 04*, of both agents can be evaluated. The mechanisms used in this work to evaluate the quality of the information of an agent are: disjunction, conjunction, certainty degree and inconsistency/sub-determination degree [2], [4], [9] and [12].

Disjunction

The disjunction operator ($\vee$) below, defined in [9], is applied when an agent needs to confirm a hypotheses or reinforce its beliefs about a certain component.

$$[\mu_1, \nu_1] \vee [\mu_2, \nu_2] = [\max(\mu_1, \mu_2), \min(\nu_1, \nu_2)] \qquad (1)$$

where, the evidential factors are:
$[\mu_1, \mu_2], [\nu_1, \nu_2] \in [0,1]$.

In the example of Fig.6, the certainty degrees of the numerical field three last figures need to be increased, because they are smaller than the certainty degrees obtained by the corresponding literal field. Therefore the *numerical agent* applies the disjunction operator on the information calculated locally and the information received from the *literal agent*, obtaining this way the following expressions:

$[0.90\ 0.11] \vee [0.89\ 0.02] \vee [0.88\ \ 0.03] = [0.90\ \ 0.02]$
$[0.80\ 0.23] \vee [0.89\ 0.02] \vee [0.88\ \ 0.03] = [0.89\ \ 0.02]$
$[0.99\ 0.40] \vee [0.89\ 0.02] \vee [0.88\ \ 0.03] = [0.99\ \ 0.02]$

The Fig.6 shows the information obtained after the application of the operator ($\vee$).

---

[3] *In the Brazilian legislation, for bank checks, the valid value is the written one.*

Conjunction

The conjunction operator ($\wedge$) below, defined in [9], is applied when an agent needs to obtain a closure value of each amount.

$$[\mu_1, \nu_1] \wedge [\mu_2, \nu_2] = [\min(\mu_1, \mu_2), \max(\nu_1, \nu_2)] \quad (2)$$

where, the evidential factors are: $[\mu_1, \mu_2]$, $[\nu_1, \nu_2] \in [0,1]$.

Information obtained after the application of the disjunction operator over the local information of the numerical value and the information received from the literal agents.	$c([\mu, \nu])$	Information obtained by the literal agent by the segmentation: σ_1, σ_2 e σ_3	$c([\mu, \nu])$
	χ_{4j}	$(\sigma_{1j}, \sigma_{2j}, \sigma_{3j})$	χ_{4j}
< 1 : [0.99 ; 0.02]>	0.97	< eleven : [0.89 ; 0.02]>	0.87
< 1 : [0.86 ; 0.01]>	0.85	< “ reais “ : [0.88 ; 0.03]>	0.85
< , : [0.90 ; 0.02]>	**0.88**		
< 0 : [0.89 ; 0.02]>	**0.87**		
< 0 : [0.99 ; 0.02]>	**0.97**		

Fig. 6. Degrees of favorable and opposite evidences, and certainty degrees after the application of disjunction operator

The conjunction operator permits to generate a unique value for σ_i and χ_i from various values σ_{ij} and χ_{ij}. In other words, a unique favorable and opposite evidential value can be obtained, as well as a unique certainty degree. For example, in the application of the operator ($\wedge$) on the *numerical* and *literal agents* local information, it is obtained:

Numerical Agent:

$[0.99\ \ 0.02] \wedge [0.86\ \ 0.01] \wedge [0.90\ \ 0.03] \wedge [0.89\ \ 0.03] \wedge [0.99\ \ 0.03] = [0.86\ \ 0.03]$

Literal Agent:

$[0.89\ \ 0.02] \wedge [0.88\ \ 0.03] = [0.88\ \ 0.03]$

This information will be sent to the analysis agent in order to interpret the evidential factors obtained for each value.

Inconsistency/Sub-determination (I/S) Degree

The calculation of the degree of I/S, defined in [4], [5], [12] allows to map in a single value the inconsistency or sub-determination of the analyzed information.

$$I/S = |\mu_1 + \nu_1 - 1| * 100 \quad (3)$$

The calculation agent does this calculation in two stages:

– application of the conjunction operator on the information received by the recognition agents, obtaining in this case:

$$[0.86\ \ 0.03] \wedge [0.88\ \ 0.03] = [0.86\ \ 0.03]$$

– the calculation for I/S is:

$$|0.86 + 0.03 - 1| * 100 = 11\%$$

This means that the obtained information – from a given check – has 11% of I/S. The acceptation or not of the check is submitted to *Rule 09*, defining a 5% limit established according to statistic calculation on a test base of Brazilian check banks.

Example 3.3:

```
Rule 09: If <I/S ∈ [0%, 5%]> then <accept check>
else <reject check>
```

Remember that the calculations above are done locally, inside each agent. This implies that the agents should be endowed with communication mechanisms. In summary, these mechanisms include three distinctive phases: (i) the settlement of a connection between agents; (ii) the solicitation and communication of determined information; (iii) the end of connection.

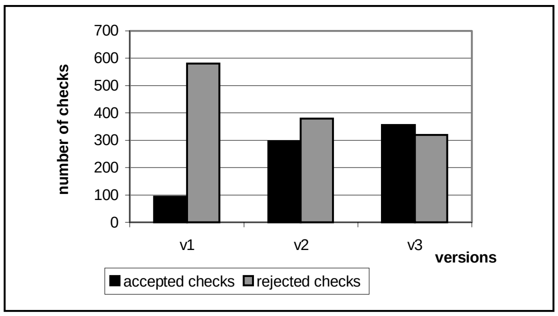
4. Other Works

In this application domain, Montoliu [6] proposes a solution for the problem of treating French bank checks, using the concept of reactive agent. In this propose, three types of agent are defined: (i) base agents, that are the classifiers (e.g. RN, PPV and HMM); (ii) macro agents, that are entities composed by agents of base regrouped by specialities (e.g. global treatment word, number treatment); (iii) meta agents, that are entities that combine the results produced by the base agents.

The main advantage of this method is the velocity in which the result is produced, because of the classificatory use in c*ascade*. In other hand, the main inconvenience is the lack of interaction between agents, that are only reactive, and in the absence of intelligence in each level of agent. This fact does not allow the agents to exchange partial results and beliefs. Beyond, that there aren’t interactions between stages of treatment, and that makes the interpretation process of the check sequential, direct and potentially less robust.

5. Results

The tests done to prove the robustness of the system were realized on three different versions of the system: (i) The *v1 test* corresponds to check analysis without any interaction between the agents; (ii) In the *v2 test* the recognition agents interact with a segmentation agent during the check analysis to request a new segmentation; (iii) The *v3 test* represents the case where all agents are able to interact;



This graphic shows that the interaction between these agents results in a highly robust treatment process, as the exchanges among the agents can resolve situations which are apparently difficult, or impossible to resolve with a unique expert.

6. Conclusion

The treatment of handwritten Brazilian bank checks is a very complex problem and it requires large computing resources to automate them. However, it's a domain where the tasks are very well defined and the tasks encapsulation in independent agents, allows a progressive development of the system, as well as the reuse of these agents in other applications. The interaction between these agents makes the process of checks treatment robust, because the agents have abilities to learn, reason and resolve conflicts. The presence of inconsistent information is frequent in the interaction between *literal* and *numerical* agents, because they have to recognize the same information, however codified in different formats. This way, to treat appropriately the inconsistency, were used some concepts and operators of paraconsistent logic allowing.

7. References

[1] Abe, J. M.; Ávila, B. C.; Prado, J. P. A., Multi-Agents and Inconsistence, In: ICCIMA'98, p.131-42, Traralgon, Austrália, 1998.

[2] Ávila, B.C.; Abe, J.M.; Prado, J.P.A., *Paralog_e: A Paraconsistent Logic Programming Language*, In: ICCIMA'98, p.143-148, Traralgon, Austrália, 1998.

[3] Brito Jr. A.S.; Bortolozzi F.; Lethelier E.; Sabourin R., *Numeral Slant Normalization Basead on Contextual Informational*, Accept on 7th International Workshop on Frontiers in Handwritten Recognition, Amsterdam, Netherlands, September, 2000.

[4] da Costa, N.C.A., *Calculs Propositionnels pour les Systèsmes Formales Inconsistants*, Compte Rendu Acad. des Sciencis, 257, pp. 3790-3792, 1963, *in French*.

[5] Enembreck, F.; Ávila B.C.; Sabourin, R., *Decision Tree-Based Paraconsistent Learning*, In: Proc. of XIX International Conference of the Chilean Computer Science Society, pp. 32-44, IEEE Computer Society Press, Talca, Chile, 1999.

[6] Montoliu L., *Architecture multi-agents et réseaux connexionnistes: Application à la lecture de chèques manuscrits*, Thèse de Doctoraty, Laboratoire d'Informatic, Unité C.N.R.S. nº 1439, Ècole polytechinique, Paris, France, 1995.

[7] Morita M.E.; Bortolozzi F.; Facon J.; Sabourin R., Mathematic Morphology and Weighted Least Square to Correcting Handwritten, Proceedings of IEEE – ICDAR99 – The 50th International Conference on Document Analysis na Recognition, Vol.I, pp. 430-433, Bangalore, India, September,1999.

[8] Oliveira, L.E.; Bortolozzi F.; Lethelier E.; Sabourin R., *A New Segmentation Approach for Handwritten* Digits, 15th International Conference on Pattern Recognition, Barcelona, Spain, September, 2000.

[9] Prado J.P.A.; Abe, J.M.; Rilo M.A., *A Paraconsistent Planning System*, In: International Conference on CAD/CAM, Robotics and Factories of the Future, 1995, p.272-283.

[10] Scalabrin E., Bortolozzi F., Kaestner C., Sabourin R., *Multicheck : Une Architecture d'agents cognitifs indépendants pour le traitement automatique des chèsques bancaires Brésiliens,* In: CIFED'98, Canada, Maio, 1998, *in French.*

[11] Scalabrin, E. E., Conception et Réalisation d'environnement de développement de systèmes d'agents cognitifs, Thèse de Doctorat, Université de Technologie de Compiègne , pp. 169, France, 1996.

[12] Subrahmanian V., S.; *Towards a Theory of Evidential Reasoning*, Logic Colloquium'87, The European Summer Meeting of the Association for Simbolic Logic, Granada, Spain, July, 1987.