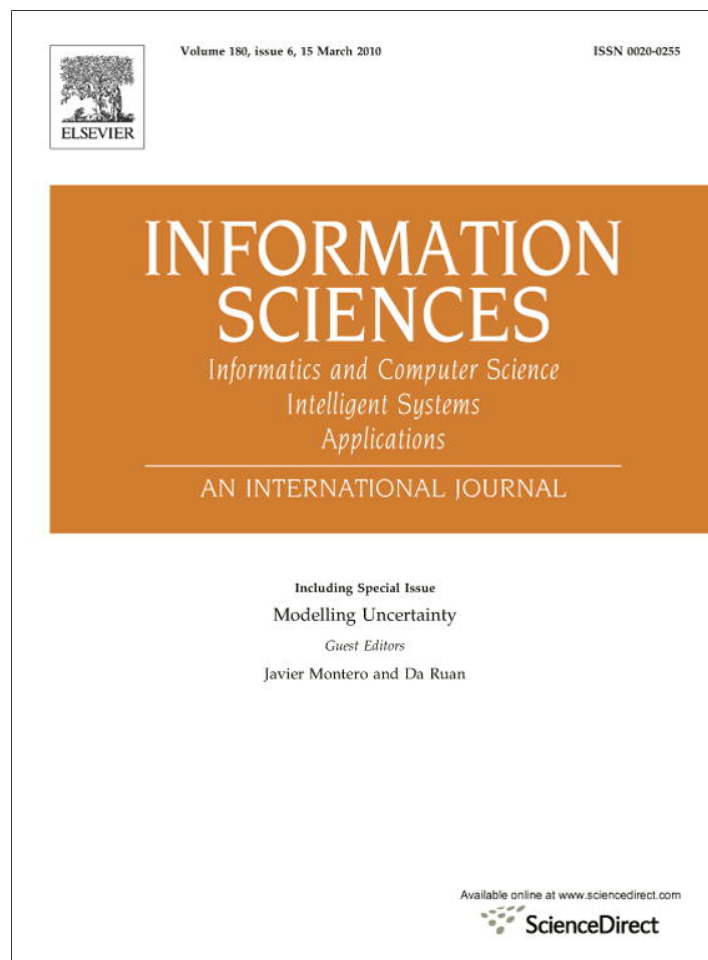


Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.



This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

<http://www.elsevier.com/copyright>



Contents lists available at ScienceDirect

Information Sciences

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

Multi-agent neural business control system

M. Lourdes Borrajo^a, Juan M. Corchado^b, Emilio S. Corchado^c, María A. Pellicer^c, Javier Bajo^{b,*}^aDepartamento de Informática, University of Vigo, Escuela Superior de Ingeniería Informática, Edificio Politécnico, Campus Universitario As Lagoas s/n, 32004 Ourense, Spain^bDepartamento de Informática y Automática, University of Salamanca, Plaza de la Merced s/n, 37008 Salamanca, Spain^cDepartamento de Ingeniería Civil, University of Burgos, Esc. Politécnica Superior, Edificio C, C/ Francisco de Vitoria, 09006 Burgos, Spain

ARTICLE INFO

Article history:

Received 3 February 2008

Received in revised form 12 June 2009

Accepted 19 November 2009

Keywords:

Agents technology

Business control system

Case-based reasoning

Maximum Likelihood Hebbian Learning

Reasoning

Experience management

ABSTRACT

Small to medium sized companies require a business control mechanism in order to monitor their modus operandi and analyse whether they are achieving their goals. A tool for the decision support process was developed based on a multi-agent system that incorporates a case-based reasoning system and automates the business control process. The case-based reasoning system automates the organization of cases and the retrieval stage by means of a Maximum Likelihood Hebbian Learning-based method, an extension of the Principal Component Analysis which groups similar cases by automatically identifying clusters in a data set in an unsupervised mode. The multi-agent system was tested with 22 small and medium sized companies in the textile sector located in the northwest of Spain during 29 months, and the results obtained have been very satisfactory.

© 2009 Elsevier Inc. All rights reserved.

1. Introduction

Small to medium sized companies require a business control mechanism in order to monitor their modus operandi and analyse whether they are achieving their goals. Such mechanisms are constructed around a series of organizational policies and specific procedures dedicated to giving reasonable guarantees to their executive bodies. This group of policies and procedures is referred to as “controls”, and conforms to the structure of the business control of the company. As a consequence of this, the need has arisen for periodic internal audits. However, evaluating and predicting the evolution of these types of business entities, which are characterized by their great dynamism, tends to be a complicated process. It is necessary to construct models that facilitate the analysis of the work carried out in changing environments such as finance.

The processes carried out inside a company are grouped into functional areas [11] denominated “Functions”. A Function is a group of coordinated and related activities that are systematically and iteratively carried out during the process of reaching the company's objectives [36]. The functions that are usually carried out within a company, as studied within the framework of this research, are: Purchases, Cash Management, Sales, Information Technology, Fixed Assets Management, Compliance to Legal Norms and Human Resources. Each one of these functions is broken down into a series of activities. For example, the Information Technology function is divided into the following activities: Computer Plan Development, Study of Systems, Installation of Systems, Treatment of Information Flows, and Security Management.

Each activity is comprised of a number of tasks. For example, the activity Computer Plan Development, which belongs to the Information Technology function, can be divided into the following tasks:

* Corresponding author. Tel.: +34 923 277100; fax: +34 923 277101.

E-mail addresses: jbajope@usal.es, jbajope@upsa.es (J. Bajo).

1. Definition of the required investment in technology in the short and medium term.
2. Coordination of the technology investment plan and the development plan for the company.
3. Periodic evaluation of the established priorities in the technology investment plan to identify their relevance.
4. Definition of a working group focused on the identification and control of the information technology policy.
5. Definition of a communication protocol, both bottom-up and top-down, to involve the company employees in the maintenance strategic plan.

Control procedures also have to be established in the tasks to ensure that the established objectives are achieved.

Decision support systems (DSSs) oriented to the concept of experience management, focus on experience processing and its corresponding management [43]. As shown in [5,43], the main stages of the system are to: discover experience, capture and collect experience, model the experience, store the experience, evaluate the new experience, adapt the experience and transform the experience into knowledge. The management of experience processing for each process stage includes analysis, planning, organisation, support and collaboration [35,44,17], and study strategies for small-medium business information systems. One of the main benefits identified from adopting strategic information systems is the obtainment of information to manage the business more efficiently and competitively, which is, in effect, the goal of DSSs. Houben et al. [25] argue for the use of a knowledge-based DSS to make strategic decisions in DSSs. These systems use symbolic methodologies, but non-symbolic decision-making methodologies such as neural nets [34] and genetic algorithms [50] are also employed in many situations. It is possible to find some examples of decision support systems in the field of textile companies. Vassileva [46] proposes an application of MKO-1 software system in the operative planning of the production program for the spinning department in a textile company. Thomassey and Fiordaliso [45] present a sales forecasting system based on clustering and decision tree classification tools that perform mid-term forecasting. Min and Liu [38] propose a test-cost-sensitive approach, and Li and Sun [36] propose the use of CBR for business failure prediction. These systems fail to take into account the dynamic characteristics of the textile sector, focusing exclusively on specific case studies.

Multi-agent systems (MAS) have become increasingly relevant for developing applications in dynamic [29], flexible environments and are specifically recommended for solving dynamic distributed problems in many fields such as e-commerce [44], supply chain management (SCM) [18], tourism recommendation [12], health care [15], oceanography [14], or shopping recommendations [2,3]. Agents can be characterized through their capacities in areas such as autonomy, communication, learning, goal orientation, mobility, persistence, etc. Autonomy, learning and reasoning are especially important aspects for an agent. These capabilities can be modelled in different ways and with different tools [49]. One of the possibilities is the use of Case-Based Reasoning (CBR) systems [1,7].

This paper presents a multi-agent architecture specifically designed to facilitate the process of internal auditing in companies. The architecture incorporates intelligent agents with learning and adaptation capabilities that facilitate the automation of auditing tasks, and are able to predict anomalous situations. These agents are the core of the architecture and incorporate a dynamic behaviour into the auditing system. In this respect, the architecture proposes an innovative tool for companies that is capable of monitoring the key parameters that characterize the company's evolution, as well as a decision support mechanism that facilitates the automatic adaptation and suggestions for evolution based on predictive algorithms. The architecture can be easily extended to different types of business, since it is highly adaptable. The firm obtains a tool to monitor its internal processes and detect inconsistent situations, and a decision support tool that reduces current needs for experts as related to existing approaches [42]. The multi-agent system developed within the framework of this investigation analyses the data that characterises each one of the activities carried out by the firm, then determines the state of each activity, calculates the associated risk, detects the inefficient processes, and generates recommendations to improve these processes. The developed model is composed of five different agent types, two of which have reasoning capabilities. One is used to identify the activities that may be improved, and the other to determine how the activities could be improved. Each of the two reasoning agents uses a different problem solving method in each of the phases of the reasoning cycle.

The developed multi-agent system is composed of two fundamental agents [6] that incorporate case-based reasoning systems:

- ISA agent (Identification of the State of the Activity) whose objectives are: to identify the state or situation of each one of the activities of the company and to calculate the risk associated with this state.
- GR agent (Generation of Recommendations), whose goal is to generate recommendations to reduce the number of inconsistent processes in the company.

Both agents are implemented using a case-based reasoning (CBR) system [1,31,33,48]. The CBR system integrated within each agent uses different problem solving techniques and shares the same case memory [26,37]. Moreover, the CBR systems proposed in the framework of this research incorporate a Maximum Likelihood Hebbian Learning (MLHL) [7] based model to automate the process of case indexing and retrieval, which may be used in problems in which the cases are predominantly characterized by numerical information. The Maximum Likelihood Hebbian Learning has been successfully applied in many other different fields, such as oceanography [13] or topology [9]. One of the aims of this research is to improve the performance of the CBR systems integrated within the ISA and GR agents by means of incorporating the MLHL into the CBR cycle stages.

Maximum Likelihood Hebbian Learning-based models were first developed as an extension of Principal Component Analysis [39,40]. The Maximum Likelihood Hebbian Learning-based method attempts to identify a small number of data points that are necessary to solve a particular problem to the required accuracy. These methods have been successfully used in the unsupervised investigation of structure in data sets [7,8]. We have previously investigated the use of artificial neural networks [17] and Kernel Principal Component Analysis (KPCA) [20,21] to identify cases that will be used in a case-based reasoning system. In this paper, we present a novel hybrid technique. The ability of the Maximum Likelihood Hebbian Learning-based methods presented in this paper to cluster cases/instances and to associate cases to clusters can be used to successfully pare down the case-base without losing valuable information.

The rest of the paper presents the Maximum Likelihood Hebbian Learning-based method and its theoretical background, after which the proposed multi-agent system is detailed. The results obtained after testing the system in 22 companies are evaluated and, finally, the conclusions are presented.

2. Maximum Likelihood Hebbian Learning-based method

The use of the Maximum Likelihood Hebbian Learning-based method was derived from the research of [8,20–22], etc. in the field of pattern recognition as an extension of Principal Component Analysis (PCA) [39,40]. The present research will first review Principal Component Analysis (PCA), which has been the most frequently reported linear operation involving unsupervised learning for data compression, and aims to find the orthogonal basis that maximises the data's variance for a given basis dimensionality. After this, we will outline, the Exploratory Projection Pursuit (EPP) theory. The research will show how the Maximum Likelihood Hebbian Learning-based method may be derived from PCA and how it might be viewed as a method for performing EPP. Finally, we shall explain why the Maximum Likelihood Hebbian Learning-based method is appropriate for these types of problems. This method is used by the CBR system so that one of the its agents can index and cluster the cases, and during the retrieval stage. The cases retrieved with the help of the MLHL method will help the ISA agent to qualify the states of the activities as described in Section 3.2.1.

2.1. Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a standard statistical technique for compressing data; it can be shown to give the best linear compression of the data in terms of the least mean square error. There are several artificial neural networks that have been shown to perform PCA e.g. [39,40]. The present research will apply a negative feedback implementation [23].

The basic PCA network is described by Eqs. (1)–(3). Given an N-dimensional input vector at time t , $\mathbf{x}(t)$, and an M-dimensional output vector, \mathbf{y} , with W_{ij} being the weight linking input j to output i . η is a learning rate. Then, the activation passing and learning is described by

Feedforward:

$$y_i = \sum_{j=1}^N W_{ij}x_j, \quad \forall i \quad (1)$$

Feedback:

$$e_j = x_j - \sum_{i=1}^M W_{ij}y_i \quad (2)$$

Change weights:

$$\Delta W_{ij} = \eta e_j y_i \quad (3)$$

This algorithm is equivalent to Oja's Subspace Algorithm [39]:

$$\Delta W_{ij} = \eta e_j y_i = \eta \left(x_j - \sum_k W_{kj} y_k \right) y_i \quad (4)$$

The PCA network not only causes the convergence of the weights but also causes the weights to converge in order to span the subspace of the Principal Components of the input data.

Exploratory Projection Pursuit (EPP) is a more recent statistical method aimed at solving the difficult problem of identifying structure in high-dimensional data. It does this by projecting the data onto a low dimensional subspace where a human expert physically searches for its structure. However, not all projections will reveal the data's structure equally well. An index that measures how "interesting" a given projection might be will be defined, followed by a representation of the data in terms of projections that maximise the index.

The first step in the pursuit of an exploratory projection is to define which indices represent interesting directions. An "interesting" structure is usually defined with respect to the fact that most projections of high-dimensional data onto arbitrary lines through most multi-dimensional data will produce almost Gaussian distributions [16]. Therefore, to identify

“interesting” features in data, it is necessary to look for those directions onto which the data-projections are as far from the Gaussian as possible.

It was shown in [29] that the use of a (non-linear) function creates an algorithm to find the values of W which maximize the function whose derivative is $f'(\cdot)$ assuming that W is an orthonormal matrix. This was applied in [23] to the network when it performed an Exploratory Projection Pursuit.

2.2. ε -Insensitive Hebbian Learning

It has been shown [49] that the nonlinear PCA rule

$$\Delta W_{ij} = \eta \left(x_j f(y_i) - f(y_i) \sum_k W_{kj} f(y_k) \right) \quad (5)$$

can be derived as an approximation to the best non-linear compression of the data. Thus, it is possible to start with a cost function

$$J(W) = 1^T E \left\{ \left(\mathbf{x} - Wf(W^T \mathbf{x}) \right)^2 \right\} \quad (6)$$

which is minimized to get the rule (5). [32] used the residual in the linear version of (6) to define a cost function of the residual

$$J = f_1(\mathbf{e}) = f_1(\mathbf{x} - W\mathbf{y}) \quad (7)$$

where $f_1 = \|\cdot\|^2$ is the (squared) Euclidean norm in the standard linear or nonlinear PCA rule. With this choice of $f_1(\cdot)$, the cost function is minimised with respect to any set of samples from the data set on the assumption that the residuals are chosen independently, and identically distributed from a standard Gaussian distribution. It is possible to show that the minimisation of J is equivalent to minimising the negative log probability of the residual, \mathbf{e} , if \mathbf{e} is Gaussian.

$$\text{Let } p(\mathbf{e}) = \frac{1}{Z} \exp(-\mathbf{e}^2) \quad (8)$$

Then, a general cost function associated with this network can be denoted as

$$J = -\log p(\mathbf{e}) = (\mathbf{e}^2) + K \quad (9)$$

where K is a constant. Therefore, performing gradient descent on J ,

$$\Delta W \propto -\frac{\partial J}{\partial W} = -\frac{\partial J}{\partial \mathbf{e}} \frac{\partial \mathbf{e}}{\partial W} \approx \mathbf{y}(2\mathbf{e})^T \quad (10)$$

where a less important term has been discarded. See [29] for more details.

In general [42], the minimisation of such a cost function may be thought to make the probability of the residuals greater depending on the probability density function (pdf) of the residuals. Thus, if the probability density function of the residuals is known, this knowledge could be used to determine the optimal cost function. [21] investigated this with the (one dimensional) function:

$$p(\mathbf{e}) = \frac{1}{2 + \varepsilon} \exp(-|\mathbf{e}|_\varepsilon) \quad (11)$$

where

$$|\mathbf{e}|_\varepsilon = \begin{cases} 0 & \forall |\mathbf{e}| < \varepsilon \\ |\mathbf{e}| - \varepsilon & \text{otherwise} \end{cases} \quad (12)$$

with ε being a small scalar ≥ 0 .

Fyfe and MacDonald [21] described this in terms of noise in the data set. However, the authors of the present research feel that it is more appropriate to state that, with this model of the pdf of the residual, the optimal $f_1(\cdot)$ function is the ε -insensitive cost function:

$$f_1(\mathbf{e}) = |\mathbf{e}|_\varepsilon \quad (13)$$

In the case of the negative feedback network, the learning rule is

$$\Delta W \propto -\frac{\partial J}{\partial W} = -\frac{\partial f_1(\mathbf{e})}{\partial \mathbf{e}} \frac{\partial \mathbf{e}}{\partial W} \quad (14)$$

which gives:

$$\Delta W_{ij} = \begin{cases} 0 & \text{if } |e_j| < \varepsilon \\ \text{otherwise} & \eta y(\text{sign}(e_j)) \end{cases} \quad (15)$$

The difference with the common Hebb learning rule is that the sign of the residual is used instead of the value of the residual [25]. Because this learning rule is insensitive to the magnitude of the input vectors \mathbf{x} , the rule is less sensitive to outliers than the usual rule based on mean squared error. This change in viewing the difference after feedback as simply a residual rather than an error allows the possibility of considering a family of cost functions for which each member is optimal for a particular probability density function associated with the residual.

2.3. Applying Maximum Likelihood Hebbian Learning

The Maximum Likelihood Hebbian Learning algorithm can now be constructed based on the previously outlined concepts. The ε -insensitive learning rule is clearly only one of a possible family of learning rules which are suggested by the family of exponential distributions. This family was called an exponential family in [27], although statisticians use this term for a somewhat different family. Let the residual after feedback have probability density function

$$p(\mathbf{e}) = \frac{1}{Z} \exp(-|\mathbf{e}|^p) \quad (16)$$

Then a general cost function associated with this network can be denoted as

$$J = E(-\log p(\mathbf{e})) = E(|\mathbf{e}|^p + K) \quad (17)$$

where K is a constant independent of W and the expectation is taken over the input data set. Therefore, performing gradient descent on J gives

$$\Delta W \propto -\frac{\partial J}{\partial W} |_{W(t-1)} = -\frac{\partial J}{\partial \mathbf{e}} \frac{\partial \mathbf{e}}{\partial W} |_{W(t-1)} \approx E\{\mathbf{y}(p|\mathbf{e}|^{p-1} \text{sign}(\mathbf{e}))^T |_{W(t-1)}\} \quad (18)$$

where T denotes the transpose of a vector, and the operation of taking powers of the norm \mathbf{e} is on an element wise basis, as it is derived from a derivative of a scalar with respect to a vector.

Computing the mean of a function of a data set (or even the sample averages) can be tedious. In an attempt to cater to the situation in which samples keep arriving, the data set was investigated and an online learning algorithm was derived. If the conditions of stochastic approximation [30] are satisfied, it may be approximated with a difference equation. Clearly, the function to be approximated is sufficiently smooth and the learning rate can be made to satisfy $\eta_k \geq 0$, $\sum_k \eta_k = \infty$, $\sum_k \eta_k^2 < \infty$ which results in the following rule:

$$\Delta W_{ij} = \eta \cdot y_i \cdot \text{sign}(e_j) |e_j|^{p-1} \quad (19)$$

The values of $p < 2$ for leptokurtotic residuals (more kurtotic than a Gaussian distribution), were expected to be appropriate, while for platykurtotic residuals (less kurtotic than a Gaussian), values of $p > 2$ were expected to be appropriate. Researchers from the community investigating Independent Component Analysis [27,28] have shown that it is less important to get the exact distribution when searching for a specific source than it is to get an approximately correct distribution i.e. all supergaussian signals can be retrieved using a generic leptokurtotic distribution and all subgaussian signals can be retrieved using a generic platykurtotic distribution. The experiments conducted in this research will tend to support this to some extent but it is often the case that accuracy and speed of convergence are improved with a greater accuracy in the choice of p . Therefore the network operation is:

Feedforward:

$$y_i = \sum_{j=1}^N W_{ij} x_j, \quad \forall i \quad (20)$$

Feedback:

$$e_j = x_j - \sum_{i=1}^M W_{ij} y_i \quad (21)$$

Weights change:

$$\Delta W_{ij} = \eta \cdot y_i \cdot \text{sign}(e_j) |e_j|^{p-1} \quad (22)$$

Fyfe and MacDonald [21] described their rule as performing a type of PCA, but this is not strictly true since only the original (Oja) ordinary Hebbian rule actually performs PCA. It might be more appropriate to link this family of learning rules to Principal Factor Analysis since PFA makes an assumption about the noise in a data set and then removes the assumed noise from the covariance structure of the data before performing a PCA. This is a similar situation in that the PCA-type rule is based on the assumed distribution of the residual. By maximising the likelihood of the residual with respect to the actual distribution, the learning rule is being matched to the probability density function of the residual.

More importantly, it is also possible to link the method to the standard statistical method of Exploratory Projection Pursuit: now the nature and quantification of the interestingness is in terms of how likely the residuals are under a particular model of the probability density function of the residuals. In the results reported later, the data is sphered before applying the learning method thus showing that with this method it is also possible to find interesting structure in the data.

2.4. Sphering of the data

Because a Gaussian distribution with mean a and variance x is no more or less interesting than a Gaussian distribution with mean b and variance y —indeed this second order structure can obscure a higher order and a more interesting structure—this information is removed from the data. This is known as “sphering”. That is, the raw data is translated until its mean is zero. It is then projected onto the principal component directions and multiplied by the inverse of the square root of its eigenvalue to give data which has mean zero and is of unit variance in all directions. So for input data X the covariance matrix is.

$$\sum = \langle (X - \langle X \rangle)(X - \langle X \rangle)^T \rangle = UDU^T \quad (23)$$

where U is the eigenvector matrix, D the diagonal matrix of eigenvalues, T denotes the transpose of the matrix, and the angled brackets indicate the ensemble average. New samples drawn from the distribution are transformed to the principal component axes to give y where

$$y_i = \frac{1}{\sqrt{D_i}} \sum_{j=1}^n U_{ij}(X_i - \langle X_i \rangle), \quad \text{for } 1 \leq i \leq m \quad (24)$$

where n is the dimensionality of the input space and m is the dimensionality of the sphered data.

3. Multi-agent business control system

This section describes the multi-agent business control system in detail. Although the aim is to develop a generic model useful in any type of small to medium company, the initial work focused on the textile sector to facilitate the research and its evaluation. The model presented here may be extended or adapted for other sectors. Twenty-two companies from the north-west of Spain, working mainly for the Spanish market, collaborated in this research. The companies have different levels of automation and all of them were very interested in a tool such as the one developed within the framework of this investigation. After analysing the data relative to the activities developed within a given firm, the constructed multi-agent system is able to determine the state of each of the activities and calculate the associated risk. It also detects any inefficient processes and generates recommendations for improving these processes. A Firm agent was assigned for each firm in order to collect new data and allow consultations. The Expert agents, as shown in Section 3.1, help the auditors and business control experts that collaborate in the project to provide information and feedback to the multiagent system. These experts generate prototypical cases from their experience and they receive assistance in developing the Store agent case-base. As shown in Fig. 1, the problem solving mechanism developed makes its decision with the help of two CBR-based agents and a Store agent, whose memory has been fed with cases constructed with information provided by the firm (through its agent) and with prototypical cases identified by 34 business control experts, using personal agents who have collaborated and supervised the developed model.

The two CBR-based agents (ISA agent-Identification of the State of the Activity and GR agent-Generation of Recommendations) incorporate a case-based reasoning system as a reasoning mechanism. The cycle of operations of each case-based reasoning system is based on the classic life cycle of a CBR system [1,47]. Both agents are communicated with the Store agent that stores the shared case base (Table 1 shows the attributes of a case), as can be seen in Fig. 1. A case represents the “shape” of a given activity developed in the company.

Every time that it is necessary to obtain a new estimate of the state of an activity, the multi-agent system evolves through several phases. On the one hand, this evolution allows the multi-agent system, to identify the latest situations most similar to the current situation in the retrieval stage, and to adapt the current knowledge in the reuse stage in order to generate an initial estimate of the state of the activity being analysed. On the other hand, it is possible to identify old situations that serve as a basis to detect the inefficient processes developed within the activity and to select the best of all possible activities. The activity selected will then serve as a guide for establishing a set of recommendations that allow the activity, its function, and the company itself, to develop in a more positive way. The retention phase guarantees that the system evolves in parallel with the firm, basing the corrective actions on the calculation of the error previously made. The following sections describe the different phases of the proposed model. The proposed MLHL method can be used to cluster data and to identify, during the retrieval stage of the CBR system, the most appropriate cases for solving a particular problem. It can also be used during adaptation to identify a final proposed solution for a given problem.

3.1. Expert agent: data acquisition

The data used to construct the model were obtained by surveys conducted with business experts in the different functional areas of various firms, using the Expert agents. This type of survey attempts to reflect the experience of the experts

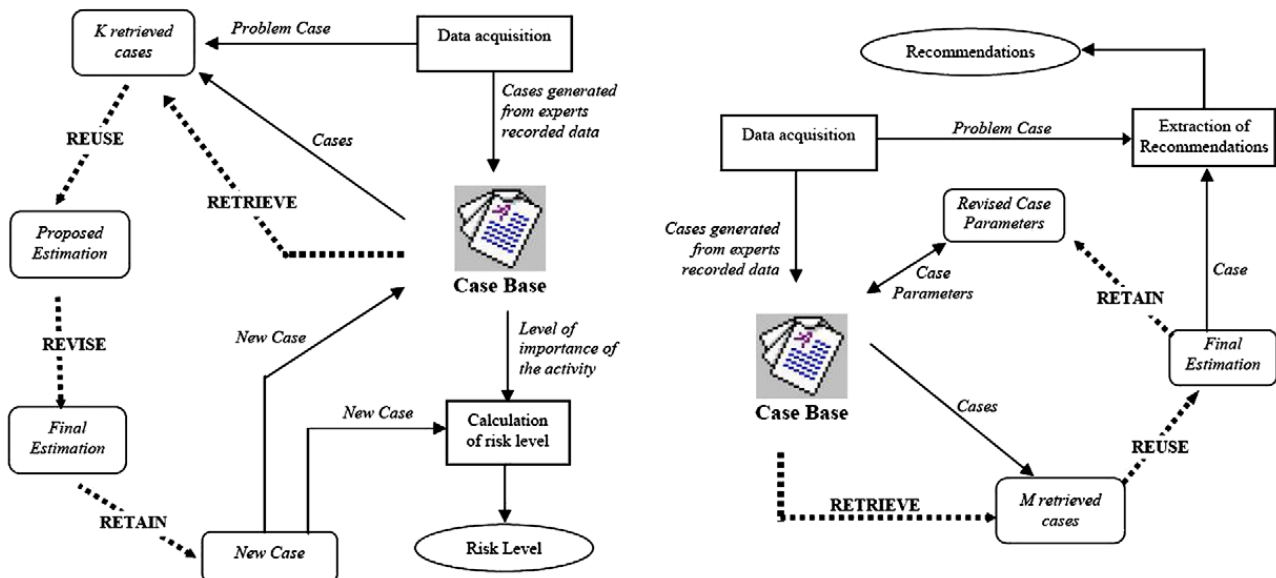
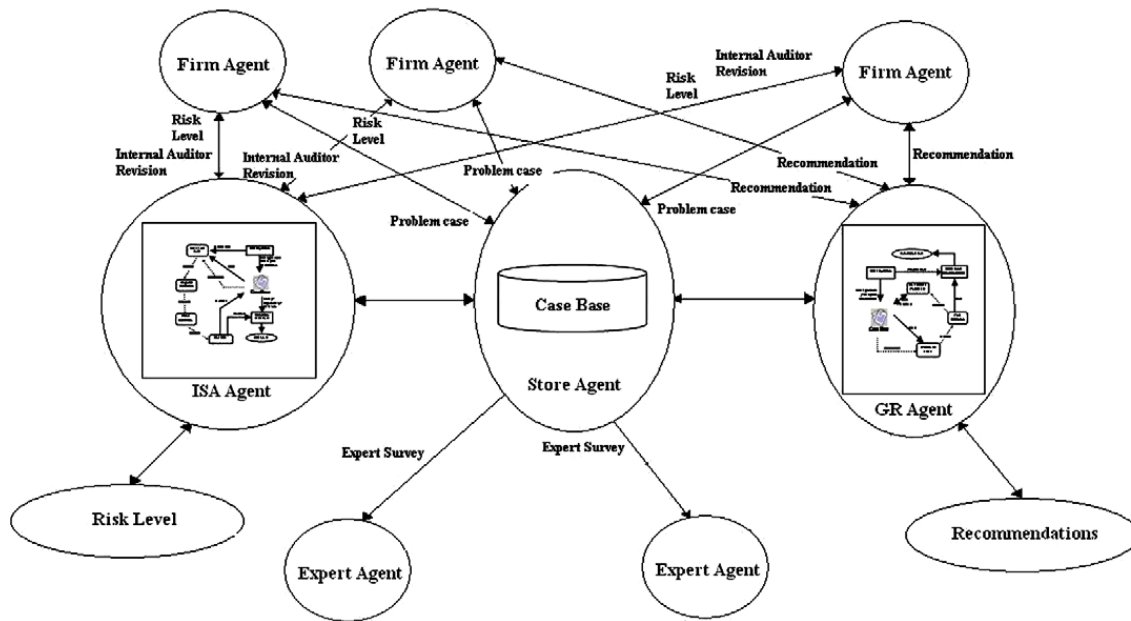


Fig. 1. Multi-agent system reasoning process. (a) Multi-agent architecture. (b) ISA agent internal structure. (c) GR agent internal structure.

Table 1
Case structure.

| Problem | Solution |
|-------------|---|
| Case number | Input vector Function number Activity number Reliability Activity State |

in their different fields. For each activity, the survey presents two possible situations: the first one tries to reflect the situation of an activity with an incorrect activity state, and the second one tries to reflect the situation of an activity with a satisfactory activity state. Both situations will be valued by a human expert using a percentage. Fig. 2 shows a generic survey relative to any activity [51]. The data acquired by means of surveys were used to build the prototype cases for the initial Store agent case base.

Table 1 shows the case structure that constitutes the case base. Each case is composed of the following attributes:

- **Case number:** Unique identification: positive integer number.
- **Input vector:** Information about the tasks (n sub-vectors) that constitute an industrial activity: $((IR_1, V_1), (IR_2, V_2), \dots, (IR_n, V_n))$ for n tasks. Each task sub-vector has the following structure (IR_i, V_i) :

Function: Function name
Activity: Activity name

IMPORTANCE RATE

| TASKS | IMPORTANCE RATE | | | | | Value of the realization state of the task to reflect the situation of an activity with an incorrect activity state | | | | | | | | | | Value of the realization state of the task to reflect the situation of an activity with a satisfactory activity state | | | | | | | | | |
|--------|---------------------|----------------|--------------------|-----------------|----------------------|---|---|---|---|---|---|---|---|---|----|---|---|---|---|---|---|---|---|---|----|
| | Very Low Importance | Low Importance | Average Importance | High Importance | Very High Importance | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Task 1 | | | | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Task 2 | | | | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Task 3 | | | | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Task 4 | | | | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Task 5 | | | | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

Activity state (degree of perfection of the development of the activity, expressed by percentage)

Fig. 2. Generic experts 511 survey.

- IR_i : importance rate for this task within the activity. It can only take one of the following values: VHI (Very high importance), HI (High Importance), AI (Average Importance), LI (Low Importance), VLI (Very low importance). To transform this attribute in a numeric value, the conversion process, shown in Table 2, was applied.
- V_i : Value of the realization state of a given task: a positive integer number (between 1 and 10).
- **Function number**: Unique identification number for each function.
- **Activity number**: Unique identification number for each activity.
- **Reliability**: Percentage of probability of success. It represents the percentage of success obtained using the case as a reference to generate recommendations, as discussed later in Sections 3.2.3 and 3.3.3.
- **Activity state**: Degree of perfection for the development of the activity, expressed by percentage. This is the solution of a problem case.

3.2. Identification of the state of the activity agent

The ISA agent (Identification of the State of the Activity) identifies the state or situation of each activity within the company and calculates the risk associated with this situation. The agent uses the data for the activity, introduced by the Firm agent, to construct the problem case. For each task that constitutes a part of the analysed activity, the problem case is composed of the value of the realization state for that task, and its level of importance within the activity (according to the internal auditor).

In this way, a problem case for an activity of n tasks, will be composed of a vector such as: $((IR_1, V_1), (IR_2, V_2), \dots, (IR_n, V_n))$. For example, Fig. 3 shows the data for the activity “Computer Plan Development” (Function “Information Technology”) introduced by the Firm agent, to construct a problem case. The vector corresponding to this problem case is: $((2,1),(4,8),(4,8),(5,9),(4,6))$.

3.2.1. Retrieval step

The ISA agent communicates with the Store agent to retrieve K cases – the cases most similar to the problem case; this is done with the proposed Maximum Likelihood Hebbian Learning method. Applying Eqs. (20)–(22) to the case base, the MLHL algorithm automatically groups the cases into clusters. The proposed indexing mechanism classifies the cases/instances

Table 2
Assessment of the different levels of importance of the tasks.

| Importance rate | Numeric value |
|----------------------------|---------------|
| VHI (Very High Importance) | 5 |
| HI (High Importance) | 4 |
| AI (Average Importance) | 3 |
| LI (Low Importance) | 2 |
| VLI (Very Low Importance) | 1 |

Function: Information Technology
Activity: Computer Plan Development

| TASKS | IMPORTANCE RATE | | | | | Value of the realization state of the task | | | | | | | | | |
|---|---------------------|----------------|--------------------|-----------------|----------------------|--|---|---|---|---|---|---|---|---|----|
| | Very Low Importance | Low Importance | Average Importance | High Importance | Very High Importance | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Definition of the required investment in technology in the short and medium time | | X | | | | X | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Coordination of the technology investment plan and the development plan of the company | | | | X | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | X | 9 | 10 |
| Periodic evaluation of the established priorities on the technology investment plan to identify their relevance | | | | X | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | X | 9 | 10 |
| Definition of a working group focused in the identification and control of the information technology policy | | | | | X | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | X | 10 |
| Definition of a communication protocol, in both directions: bottom-up and top-down, to involve the firm employees in the maintenance strategic plan | | | | X | | 1 | 2 | 3 | 4 | 5 | X | 7 | 8 | 9 | 10 |

Fig. 3. Example of the data of a problem case.

automatically, clustering together those with a similar structure. This technique attempts to find interesting low dimensional projections within the data so that humans can investigate the structure of the data by eye. One of the great advantages of this technique is that it is an unsupervised method, so it is not necessary to have any information about the data beforehand. When a new problem case is presented to the CBR system, it is identified as belonging to a particular type by once again applying Eqs. (20)–(22). This mechanism may be used as a universal retrieval and indexing mechanism to be applied to any problem similar to that presented here.

Fig. 4 shows the pseudocode for this retrieval phase. X represents the set of cases that introduces information about an activity, v_p represents the vector of characteristics (attributes) that describes a new problem, P represents the set of clusters, and K is the set of retrieved cases.

Maximum Likelihood Hebbian Learning techniques are used because of the size of their database and the need to group the most similar cases together in order to help retrieve the cases that most resemble the given problem. The techniques are especially interesting for non-linear or ill-defined problems, making it possible to treat tasks involved in the processing of massive quantities of redundant or imprecise information more effectively.

3.2.2. Re-use step

This phase aims to obtain an initial estimate of the state of the activity analysed. In order to obtain this estimate, RBF networks are used [7,10,17,19]. As in the previous phase, the number of attributes of the problem case depends on the activity analysed. Therefore, it is necessary to establish one RBF network system for each of the activities to be analysed.

The K cases retrieved in the previous phase are used by the RBF network as a training group that allows it to adapt its configuration to the new problem encountered before generating the initial estimation.

The topology of each of the RBF networks used in this task consists of: an input layer with as many neurons as attributes possessed by the input vector that constitutes the problem descriptor $((IR_1, V_1), (IR_2, V_2), \dots, (IR_n, V_n))$; a hidden layer with 14

```

procedure retrieve_ISA (input:  $X, v_p$ ; output:  $K$ )
{
00   begin
01     /*obtain_clusters:  $P \leftarrow MLHL(X)$ */
02     for each case  $c \in X$  do
03       retrain_network: error  $\leftarrow$  ann( $c$ )
04       modify_weights: ann.learn( $c$ )
05     get_cases:  $K \leftarrow MLHL(v_p)$ 
06   end.
}
    
```

Fig. 4. Pseudocode of the retrieval phase of the ISA agent.

centres; and an output layer with a single neuron corresponding to the variable to be estimated (correction level or state of activity analysed in x percent).

Fig. 5 shows the pseudocode of the algorithm that roughly illustrates the steps that need to be followed in order to obtain an initial estimate, using both the K cases retrieved in the previous phase, and the descriptor of the problem for which an estimate needs to be made. In the algorithm, v_p represents the vector of characteristics (attributes) that form the problem case, K is the group of most relevant retrieved cases, $confRBF$ is the group of neurons that make up the topology of the RBF network and s_i represents the initial solution generated for the current problem.

The RBF network is characterized by its ability to adapt, to learn rapidly, and to generalize. Within this system the network specifically acts as a mechanism capable of absorbing knowledge about a certain number of cases and generalizing from them. During this process, the RBF network, interpolates and carries out predictions without forgetting any part of those already carried out. The Store agent acts as a permanent memory capable of maintaining many cases or experiences while the RBF network, which is in the ISA agent, acts as a short term memory that is able to recognize recently learnt patterns and to generalize from them.

3.2.3. Revision step

The objective of the revision phase is to confirm or refute the initial solution proposed by the RBF network, thereby obtaining a final solution and calculating the control risk.

In view of the initial estimation or solution generated by the RBF network, the internal auditor (through the Firm's agent) will be responsible for deciding if the solution is accepted. For this reason the Firm's agent is based on the knowledge retained by the internal auditor, specifically, knowledge about the company with which the auditor is working. If the auditor considers that the estimation given is valid, the system will accept the solution as the final solution and in the following phase of the CBR cycle, a new case will be stored in the Store agent case base consisting of the problem case and the final solution. The system will assign the case an initial reliability of 100%. However, if the internal auditor considers the solution given by the system to be invalid, he or she will propose a solution which the system will accept as the final solution and which, together with the problem case, will form the new case to be stored by the Store agent in the following phase. This new case will be given a reliability of 30%. This value was decided by various auditors who determined that assigning a reliability of 30% would be up to the internal auditor.

The ISA agent calculates the control risk associated with the activity from the final solution. Every activity developed in the business sector has a risk associated with it that indicates the negative influence that affects the optimal operation of the firm. In other words, the control risk of an activity measures the impact that the current state of the activity has on the business process as a whole. In this study, the level of risk is valued at three levels: low, medium and high. The calculation of the level of control risk associated with an activity is based on the current state of the activity and its level of importance. This latter value was obtained after analysing data obtained from a series of questionnaires (98 in total) carried out by auditors throughout Spain. In these questionnaires the auditors were asked to rate subjects on a scale of 1–10 according to the importance or weight of each activity in terms of the function that it belonged to: the higher the importance of the activity, the greater its weight within the business control system.

The level of control risk was then calculated from the level of importance given to the activity by the auditors, and the final solution was obtained after the revision phase. For this purpose, if-then rules are employed. These rules follow the pattern shown in Fig. 6, where "auditors_importance" indicates the average level of importance assigned to the previously mentioned activity by the auditors and "activity_state" is the final solution or state of the activity.

3.2.4. Retention step

The last phase executed by the ISA (Identification of the State of the Activity) agent is the communication and incorporation of the system's memory, which is managed by the Store agent, of what has been learnt after resolving a new problem.

```

procedure reuse_ISA (input:  $v_p$ ,  $K$ ,  $confRBF$ ; output:  $s_i$ )
{
00  begin.
01  while TRUE do /* infinite loop */
02      for each case  $c \in K$  do /* network adaptation using  $K$  cases */
03          retrain_network: error  $\leftarrow$  annRBF( $c$ )
04          move_centers: annRBF.moveCenters( $c$ )
05          modify_weights: annRBF.learn( $c$ ) /* delta rule */
06          if (error /  $|K|$ ) < error_threshold then
07              go_to_line 8 /* end of infinite loop and adaptation */
08  generate_initial_solution:  $s_i \leftarrow$  annRBF( $v_p$ )
09  end.
}

```

Fig. 5. Pseudocode of the reuse phase of the ISA (Identification of the State of the Activity) agent.

```

if auditors importance = v1 then
  if activity_state >= v1 then
    risk_level = low
  else if activity_state < (v1-1) then
    risk_level = high
  else risk_level = average

```

Fig. 6. If-then rules pattern.

Once the revision phase has been completed and the final solution has been obtained, a new case (*problem + solution*) is constructed, which is stored in the Store agent memory. In addition to the overall knowledge update involving the insertion of a new case within the Store agent memory, the hybrid multi-agent system presented in this research carries out a local adaptation of the knowledge structures that it uses.

The Maximum Likelihood Hebbian Learning technique contained within the prototypes that are related to the activity corresponding to the new case will reorganize in order to respond to the appearance of this new case. In doing so, it will modify its internal structure and adapt itself to the new knowledge available.

The RBF network can then use the new case to carry out a complete learning cycle, updating the position of its centres and modifying the value of the weights that connect the hidden layer with the output layer.

3.3. Generation of recommendations agent

The objective of this agent is to carry out recommendations to help the internal auditor decide which actions to take, once the stages of the ISA agent have concluded, in order to improve the company's internal and external processes. This agent is totally dependent on the previous agent as it begins its work from the case (*problem + solution*) generated in the ISA – Identification of the State of Activity – agent (see Fig. 1).

3.3.1. Retrieval step

The GR-generation of recommendations agent is used to generate recommendations that can guide the internal auditor in the task of deciding the actions to be taken in order to improve the state of the activity analysed. In order to recommend changes in the execution of the business processes, it is necessary to compare the current situation in the activity, represented by the *problem case + solution*, generated by the ISA (Identification of the State of the Activity) agent, with the cases from the case base, managed by the Store agent, that best reflect the business management.

To this end, only the cases most similar to the problem case are worked on. Given that the cluster whose cases were closest to the case problem was identified during the retrieval phase of the ISA (Identification of the State of the Activity) agent, the cases of this cluster will be used in the next reuse phase. The GR agent communicates with the Store agent, and the process followed in this retrieval phase is based on the use of *query relaxation* [24] so that the cases retrieved from the case base initially meet the following conditions:

1. The solution or state of activity must be 15–20% greater than the final solution generated by the ISA agent. If enough cases are not retrieved (25 is considered sufficient) the percentages are relaxed further, increasing in range by 5%.
2. The cases should possess a level of reliability of over 50%. This constant value was established by the auditors.

Fig. 7 shows the retrieval process that was adopted, where X stands for the case group which represents the knowledge of a determined activity that exists within the memory of the Store agent, v_p represents the vector of attributes that describes the problem case, s_f is the final solution generated in the ISA agent as a solution to the problem case, s_i is the solution to case i and K is the set of the most relevant retrieved cases.

3.3.2. Re-use step

Given that the objective of this agent is to generate a series of recommendations from the problem case, it is necessary to search for a case, or combination of various cases, from the Store agent case base which can serve as a guide to generate recommendations, comparing that/those case/s with the problem case. The comparison will allow the company to detect which processes need to be modified – in other words, which tasks need to be improved.

As previously explained, the cases obtained in the retrieval phase are those which reflect the most favourable state of the activity when compared to the state presented by the activity being analysed. During the adaptation phase, the GR agent should select which among all of the cases maximises the value of each of the tasks (V_i) taking into account the level of importance (IR_i) or weight that each task has for the overall activity. This way, the problem of selecting a case from all those that have been retrieved can be structured in a way similar to a multi-criteria decision-making problem where the alternatives are the different cases retrieved, and the objective is to maximise the values of these tasks (which will then represent the attributes).

```

procedure retrieve_GR (input:  $X, v_p, s_f, m, U$ ; output:  $K$ )
{
00   begin
01     increment  $\leftarrow 0$ ;
02     repeat
03       for each case  $i \in X$  do
04         dif  $\leftarrow s_i - s_f$ 
05         if (dif  $\geq 0.15$  and dif  $\leq (0.20 + \text{increment})$ )
           and ( $u_{mi} > 0.75$ )
           and (reliability( $i$ )  $> 0.5$ ) then
06            $K \leftarrow K + i$ 
07         increment  $\leftarrow \text{increment} + 0,5$ ;
08     until  $|K| > 25$ ;
09   end
}

```

Fig. 7. Pseudocode of the retrieval phase in the generation of recommendations agent.

In this study, the initial version of the Electre method [4,41] was used in order to tackle the problem of choosing one of the alternatives. The Electre method proposes a strategy for reducing the size of all the possible solutions by segregating the most favourable case group from another group that encapsulates the least favourable cases. The application of such a method will produce the selection of one or various cases from among those retrieved.

The Electre method is based on the fact that the vector of preferential weightings subjectively associated with each attribute has been ascertained. As in this study, the weight of an attribute (represented by its level of importance) is different for each alternative, and it is necessary to obtain a single weighting vector for the attributes of the group of alternatives or retrieved cases. In this case, the weighting vector is obtained by calculating the median weight for the attribute in question, for each of the different alternatives.

Electre returns the best alternative, or group of alternatives, as a solution in the event that there is no single prevalent alternative. Given that the generation of recommendations needs to begin with a single alternative, while the output of a multicriteria decision method gives various alternatives, we will use a combination of the alternatives will be used, taking the median value for each attribute.

Fig. 8 shows the pseudocode for the reuse phase where K is the group of the most relevant cases retrieved in the previous phase, v_{el} is the case or alternative obtained after the adaptation phase from the group of cases K , $v_{el(j)}$ is the value of the attribute j of the case v_{el} , and C is the group of alternatives or cases obtained as output by the Electre method.

The case obtained as a result of the Electre method represents either the objective to be reached for the activity analysed, the standard to be followed in order to meet the objectives of the company, or more specifically, the objective associated with the activity. As such, the recommendations, which are generated retrospectively, will be used to ensure that the various tasks that make up the problem case achieve a situation which is as similar as possible to the case obtained at the output of the Electre method.

In order to generate recommendations, the output from the Electre method is compared to the problem case, comparing the values (V_i) from each of the attributes or tasks in each case. The objective is to detect which tasks should be improved, establishing an order of priorities in terms of weighting (IR_i) each task with respect to the overall weight of the activity. In other words, the output should help to identify the possible deviations of the activity and appreciate the extent of deviations in terms of the level of importance (IR_i) for each task. This way, the system generates recommendations related to the incon-

```

procedure reuse_GR (input:  $K$ ; output:  $v_{el}$ )
{
00   begin
01      $(C, n) \leftarrow \text{ELECTRE}(K)$ 
02     if  $n > 1$  then
03       for each attribute  $j$  do
04          $v_{el_j} \leftarrow \frac{\sum_{i=1}^n c_{ij}}{n}$ 
05   end
}

```

Fig. 8. Pseudocode of the reuse phase of the GR (Generation of Recommendations) agent.

sistent processes found, i.e., the differences between the values of the attributes in the problem case and those in the objective case (considered to be the standard) obtained by the Electre method, which represent the potential recommendations made by the auditor.

The group of attributes of stored cases in the Store agent case base represents the overall values that experts in each activity have communicated by means of the Expert agent. Since the characteristics of the current case (problem) are similar to the objective case obtained, the internal auditor can argue that the values of the attributes must also be similar. This provides a more convincing argument than one simply based on probabilities and estimated losses or risks.

The control recommendations that are generated by comparing the values of the current case with those of past cases also eliminate other problems such as the lack of outputs or pre-defined results. Many possible values exist as well as a large number of combinations that could be included in the recommendations made by the auditor. But not all the combinations are valid; indeed, some combinations may not even be feasible or make sense. In contrast to the CBR-based agents, both the expert systems and the neuron networks will need to have had possible outputs specified for them previously.

Based on the predictions and recommendations generated by the multi-agent system, the internal auditor may inform the company of inconsistent processes and the measures that should be adopted to resolve them. This is a decision support system that facilitates the auditing process for internal auditors.

3.3.3. Retention step

After the time required for correcting the errors that have been detected, the firm is evaluated again. Auditing experts consider that three months are enough to allow the company to evolve towards a more favourable state. If it is possible to verify that the inefficient processes and the level of risk have diminished, the retention phase is carried out, modifying the case used to generate the recommendations. The reliability (percentage of successful identifications) of this case is thereby increased by 10%. In contrast, if by chance the firm has not evolved to a better state, the reliability of the case is decreased to 10%. Furthermore, those cases with a level of reliability less than 15% are eliminated, and the remaining cases are regrouped into clusters.

4. Results and conclusions

The developed system was tested in 22 small to medium companies (12 medium sized and 10 small) in the textile sector, located in the northwest of Spain, over a period of 29 months. The data employed to generate the prototype cases, used to construct the memory of cases for the store agent, were obtained after surveying 98 auditors from Spain and 34 experts in different functional areas of the firms within the sector.

Various complete operation cycles were carried out in order to fully test the system. Each of the activities for a given company was evaluated, a level of risk was obtained, and recommendations were generated. These recommendations were communicated to the company's internal auditor, who was given a period of three months to elaborate and apply an action plan based on the aforementioned mentioned recommendations. The main objective of each of these action plans was to reduce the number of inconsistent processes within the company. New analyses were performed every three months so that the results could be recorded, compared and refined.

The data obtained demonstrates that the application of multi-agent system-generated recommendations caused a positive evolution in all firms. This evolution was reflected in the reduction of inefficient processes. The indicator used to determine the positive evolution of the companies was the state of each of the activities analysed. After analysing one of the company's activities, it was necessary to prove that the state of the activity (valued between 1 and 100) had increased beyond the state obtained in the previous three month period. Thus, it was possible to conclude that the inefficient processes had been reduced within the same activity. When there was measurable improvement in the majority of activities (above all those most relevant to the company), it could be affirmed that the company had improved its situation.

In order to reflect as reliably as possible the suitability of the system for resolving the problem of inefficient processes, the results from the analyses obtained from 22 companies were compared with those of 5 companies in which the recommendations generated by the system were not applied. In these five companies, the activities were analysed from the beginning of the three month period until the end, using the ISA agent. The recommendations generated by the second agent were not presented to the firms' managers (and consequently, the recommendations were not applied).

Before putting the system into operation in a real company, sample tests were carried out to demonstrate the correct operation of the ISA agent. Each of the activities of a given company was evaluated by the system, producing a level of risk. We requested six external and independent auditors to analyse the situation of each company. The mission of the auditors was to estimate the state of each activity, as it is the mission of the proposed system. We then compared the result of the evaluation obtained by the auditors with the result obtained by the system. The results obtained by the system are very similar to those obtained by the external auditors. Fig. 9 shows the differences between the results obtained by the system and the external auditors for the "Sales" function. In general, it could be said that these results demonstrate the suitability of the techniques used for their integration in the multi-agent system.

In order to analyse the results obtained, it should be noted that some of the recommendations implied costs that the companies were not able to afford, or involved a long term implementation. Therefore, companies are considered to have followed the recommendations if they were applied at a rate greater than 70%.

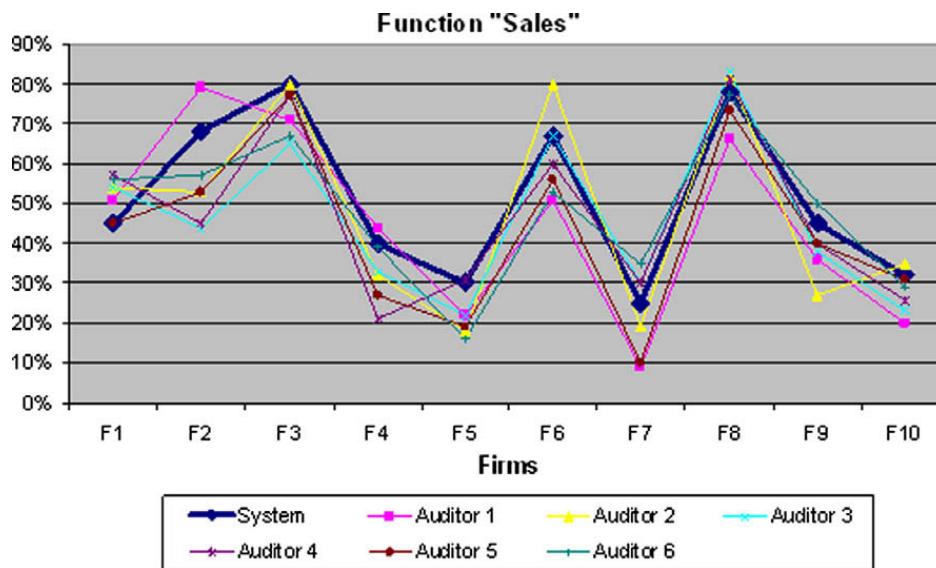


Fig. 9. Results for the sales function before and after the implementation of the multi-agent system.

The results obtained were as follows:

1. Among the companies analysed, those in which the recommendations generated by the system were applied (the results are listed in Table 3 and Fig. 10) showed that:
 - In 77% of these companies, the number of inconsistent processes was reduced improving the state of activities by an average of 15.70%.
 - In 18% of these companies, the state of activities showed an average improvement of 0.82%. In other words, the application of the recommendations generated by the system did not have any effect on the activities of the company. After studying the possible reasons for these results, it was determined that the recommendations given were not followed precisely, since only certain measures were applied while the majority of the recommendations were ignored.
 - In 5% of these companies, the inconsistent processes increased, which is to say that the application of recommendations generated by the system was contrary to the positive evolution of the company. Once the situation in the company had been analysed, it was concluded that there was a high level of disorganisation, without a clearly defined set of objectives. This means that any attempt to change the business organisation actually would lead to a worse situation.

Table 3
Improvement percentage of the analysed companies.

| | |
|---------|--------|
| Firm 1 | 13.99% |
| Firm 2 | 16.92% |
| Firm 3 | 14.48% |
| Firm 4 | 14.61% |
| Firm 5 | 14.65% |
| Firm 6 | 14.53% |
| Firm 7 | 14.53% |
| Firm 8 | 0.36% |
| Firm 9 | 0.93% |
| Firm 10 | 0.86% |
| Firm 11 | 16.99% |
| Firm 12 | –3.64% |
| Firm 13 | 16.92% |
| Firm 14 | 14.44% |
| Firm 15 | 14.59% |
| Firm 16 | 16.20% |
| Firm 17 | 16.42% |
| Firm 18 | 16.28% |
| Firm 19 | 17.13% |
| Firm 20 | 17.13% |
| Firm 21 | 17.28% |
| Firm 22 | 1.71% |

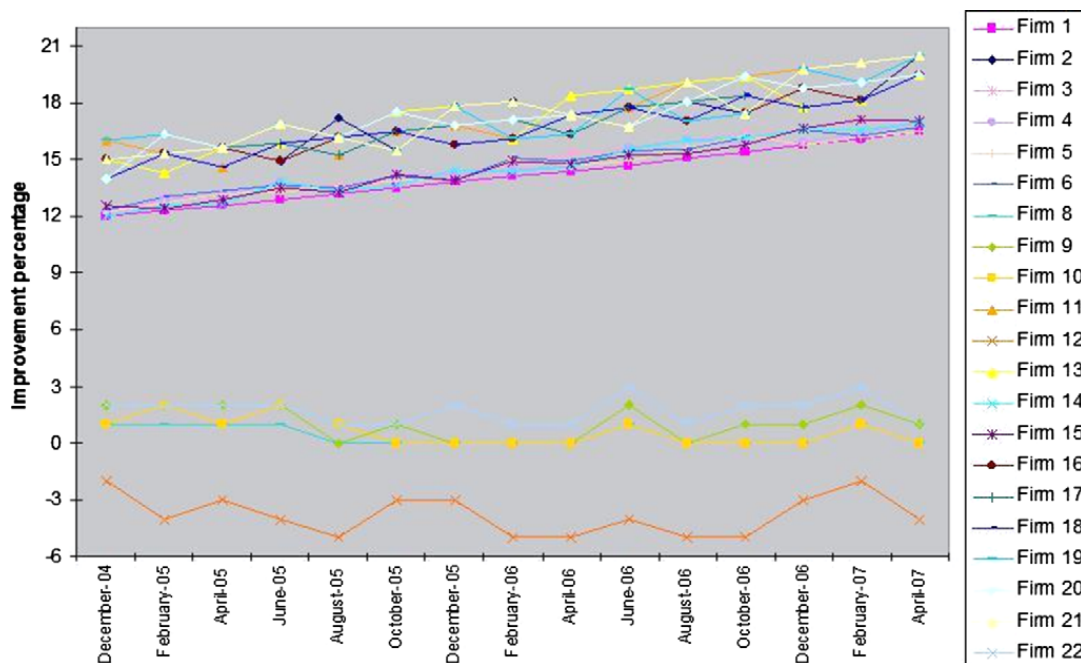


Fig. 10. Firm's evolution.

In general, it could be said that these results demonstrate the suitability of the techniques used for the integration of the developed intelligent control system. The best results occurred in the smaller sized companies. This is due to the fact that these firms have a greater facility to adopt and adapt to the changes suggested by the system's recommendations.

- For the 5 companies in which the recommendations generated by the system were not applied, the results were as follows: the companies improved their results, though reaching an average productivity that was 9% below the same measurement for other companies that did use the system.

This article presents a multi-agent system that uses two CBR systems employed as the basis for the hybridization of a multicriteria decision-making method, a Maximum Likelihood Hebbian Learning technique, and a RBF net. As such, the model developed combines the complementary properties of connectionist methods with the symbolic methods of Artificial Intelligence.

The reasoning model used can be applied in situations that satisfy the following conditions:

- Each problem can be represented in the form of a vector of quantified values.
- The case base should be representative of the total spectrum of the problem.
- Cases must be updated periodically.
- Enough cases should exist to train the network.

The prototype cases used for the construction of the case base of the Store agent are fictitious and were created from surveys carried out with auditors and experts in different functional areas. The system is able to estimate or identify the state of the activities of the firm and their associated risk. Furthermore, the system generates recommendations that will guide the internal auditor in the elaboration of action plans to improve the processes in the firm.

The complexity and dynamism of the corporate environments makes it difficult to forecast changes and possibilities. However, the developed model is able to estimate the state of the firm with precision, and propose solutions that make it possible to improve each phase of a business process. The system will produce better results if it is fed with cases related to the sector in which it will be used. This is due to the dependence that exists between the processes in the company and the sector in which it is located. Future experiments will help to identify how the constructed prototype will perform in other sectors and how it will have to be modified in order to improve its performance. We have demonstrated a new technique for case indexing and retrieval, which could be used to construct case-based reasoning systems. The basis of the method is a Maximum Likelihood Hebbian Learning algorithm. This method provides us with a very robust model for indexing the data and retrieving instances without any need of information about the structure of the data set.

We had certain problems implementing the system, partly because the management and experts were not familiar with the use of computational devices, so some courses were given to introduce them to these technologies and teach them how to use the system interface. The proposed multi-agent system has been improved to be able to provide dynamic suggestions. In this sense it is a unique system useful for dynamic environments and open enough to be used in other business environments.

Management are the most reticent about trusting the system for several reasons: (i) they do not trust the partiality of the system, and are reluctant to facilitate their internal data, and (ii) updating the information about the firm requires specialised human resources and time. However, the auditors and experts believe that the CBR–BDI agents may favour their work and provide a highly appreciated decision support tool. They believe that the multi-agent architecture has more advantages than disadvantages and that the system helped them to detect inconsistent processes in the businesses. They tend to argue that the multi-agent architecture should incorporate a shared memory of cases to compare data from different firms, but with the guarantee of data privacy.

Although the defined model has not been tested in big firms, we believe that it could work adequately, although changes would take place more slowly than in small and medium firms. We are moving in this direction and expect that an evaluation of the system will soon be possible in a major international company from the textile sector. Moreover, additional work is still required to improve the adaptation of the CBR–BDI agents when the size of the memory of cases is very high. That is our next challenge.

Acknowledgement

This work has been partially supported by the MCYT TIN2006-14630-C03-03 and the JCYL-2002-05 project SA104A05.

References

- [1] A. Aamodt, E. Plaza, Case-based reasoning: foundational issues methodological variations, and system approaches, *AI Communications* 7 (1) (1994).
- [2] J. Bajo, Y. de Paz, J.F. de Paz, Q. Martín, J.M. Corchado, SMas: a shopping mall multiagent systems, in: E.S. Corchado, H. Yin, V. Botti, C. Fyfe (Eds.), *Proceedings of the 7th International Conference Intelligent Data and Automated Learning, IDEAL'06*, LNAI, vol. 4224, Springer-Verlag, 2006, pp. 1166–1173.
- [3] J. Bajo, J.M. Corchado, L.F. Castillo, Running agents in mobile devices, in: J.S. Sichman, H. Coelho, S. Oliveira (Eds.), *Proceedings of the 10th International Conference Ibero-American Conference on AI, IBERAMIA'06*, LNAI, vol. 4140, Springer-Verlag, 2006, pp. 58–67.
- [4] S. Barba-Romero, J. Pomerai, *Decisiones Multicriterio, Fundamentos teóricos y Utilización Práctica*, Universidad de Alcalá, Alcalá de Henares, Spain, 1997.
- [5] R. Bergmann, *Experience management: foundations, development methodology and internet-based applications*, LNAI, vol. 2432, Springer, Berlin, 2002.
- [6] L. Borrajo, *Sistema híbrido inteligente aplicado a la auditoría de los sistemas internos*, Ph.D. Thesis, Universidade de Vigo, Spain, 2003.
- [7] C.J. Butz, S. Hua, J. Chen, H. Yao, A simple graphical approach for understanding probabilistic inference in Bayesian networks, *Information Sciences* 179 (6) (2009) 699–716.
- [8] E. Corchado, D. MacDonald, C. Fyfe, Optimal projections of high dimensional data, in: *Proceedings of the 2002 IEEE International Conference on Data Mining, ICDM '02*, IEEE Computer Society, 2002, pp. 589–596.
- [9] E.S. Corchado, J.M. Corchado, J. Aiken, IBR retrieval method based on topology preserving mappings, *Journal of Experimental Theory and Artificial Intelligent* 16 (3) (2004) 145–160.
- [10] J.M. Corchado, F. Díaz, L. Borrajo, F. Fdez-Riverola, *Redes Neuronales Artificiales: Un enfoque práctico*, Departamento de Publicaciones de la Universidad de Vigo, Spain, 2000.
- [11] J.M. Corchado, L. Borrajo, M.A. Pellicer, J.C. Yáñez, Neuro-symbolic system for business internal control, in: *Proceedings of the 4th Industrial Conference on Data Mining, ICDM'04, LNCS*, vol. 3275, Springer-Verlag, Berlin, 2004, pp. 1–10.
- [12] J.M. Corchado, J. Pavón, E.S. Corchado, L.F. Castillo, Development of CBR–BDI agents: a tourist guide application, in: P. Funk, P.A. Gonzalez-Calero (Eds.), *Proceedings of the 7th European Conference Advances in Case-Based Reasoning, ECCBR'04*, LNAI, vol. 3155, Springer-Verlag, Berlin, 2005, pp. 547–559.
- [13] J.M. Corchado, J. Aiken, E.S. Corchado, Florentino Fdez-Riverola, Evaluating the air–sea interactions and fluxes using an instance-based reasoning system, *AI Communications* 18 (4) (2005) 247–256.
- [14] J.M. Corchado, J. Aiken, J. Bajo, A CBP agent for monitoring the CO₂ exchange rate, book chapter in *case-based reasoning on signals and images*, Springer-Verlag, *Series on Computational Science* 73 (2007) 213–246.
- [15] J.M. Corchado, J. Bajo, Y. de Paz, D.I. Tapia, Intelligent environment for monitoring alzheimer patients, agent technology for health care, *Decision Support Systems* 44 (2) (2008) 382–396.
- [16] P. Diaconis, D. Freedman, Asymptotics of graphical projections, *The Annals of Statistics* 12 (3) (1984) 793–815.
- [17] Y. Du, C. Jiang, MengChu Zhou, You Fu, Modeling and monitoring of E-commerce workflows, *Information Sciences* 179 (7) (2009) 995–1006.
- [18] G. Finnie, J. Barker, Z. Sun, A multi-agent model for cooperation and negotiation in supply networks, in: *Americans Conference on Information Systems*, New York, USA, 2004.
- [19] B. Fritzke, Fast learning with incremental RBF networks, *Neural Processing Letters* 1 (1) (1994) 2–5.
- [20] C. Fyfe, E. Corchado, Maximum likelihood Hebbian rules, in: M. Verleysen (Ed.), *Proceedings of the 10th European Symposium on Artificial Neural Networks, ESANN'02*, Bruges, 2002, pp. 143–148.
- [21] C. Fyfe, D. MacDonald, ϵ -Insensitive Hebbian learning, *Neurocomputing* 47 (1–4) (2002) 35–57.
- [22] C. Fyfe, E. Corchado, A new neural implementation of exploratory projection pursuit, in: H. Yin, N.M. Allison, R.T. Freeman, J.A. Keane, S.J. Hubbard (Eds.), *Proceedings of the 3rd International Conference on Intelligent Data Engineering and Automated Learning, IDEAL'02, LNCS*, vol. 2412, Springer, Berlin, 2002, pp. 12–14.
- [23] C. Fyfe, R. Baddeley, Non-linear data structure extraction using simple Hebbian networks, *Biological Cybernetics* 72 (6) (1995) 533–541.
- [24] D. Gardingen, I. Watson, A web based case-based reasoning system for HVAC sales support, *Applications and Innovations in Expert Systems*, vol. VI, Springer, 1998, pp. 11–23.
- [25] F. Han, Q.H. Ling, D.S. Huang, Modified constrained learning algorithms incorporating additional functional constraints into neural networks, *Information Sciences* 178 (3) (2008) 907–919.
- [26] J. Hunt, R. Miles, Hybrid case-based reasoning, *The Knowledge Engineering Review* 9 (4) (1994) 383–397.
- [27] A. Hyvärinen, J. Karhunen, E. Oja, *Independent Component Analysis*, Wiley, 2002.
- [28] A. Hyvärinen, Complexity pursuit: separating interesting components from time series, *Neural Computation* 13 (2001) 883–898.
- [29] S. Ilarri, E. Mena, A. Illarramendi, Using cooperative mobile agents to monitor distributed and dynamic environments, *Information Sciences* 178 (9) (2008) 2105–2127.
- [30] R.L. Kashyap, C.C. Blaydon, K.S. Fu, *Stochastic Approximation, A Prelude to Neural Networks: Adaptive and Learning Systems*, Prentice-Hall, 1994.
- [31] J. Kolodner, *Case-Based Reasoning*, Morgan Kaufman, San Mateo CA, 1993.
- [32] P.L. Lai, D. Charles, C. Fyfe, Seeking independence using biologically inspired artificial neural networks, in: M.A. Girolami (Ed.), *Developments in Artificial Neural Network Theory: Independent Component Analysis and Blind Source Separation*, Springer-Verlag, 2000.
- [33] J.H. Lee, S.H. Ha, Recognizing yield patterns through hybrid applications of machine learning techniques, *Information Sciences* 179 (6) (2009) 844–850.

- [34] Y.W. Leung, J.Y. Mao, Providing embedded proactive task support for diagnostic jobs: a neural network-based approach, *Expert Systems with Applications* 25 (2) (2003) 255–267.
- [35] M. Levy, P. Powell, Information systems strategy for small and medium sized enterprises: an organisational perspective, *The Journal of Strategic Information Systems* 9 (1) (2000) 63–84.
- [36] H. Li, J. Sun, Gaussian case-based reasoning for business failure prediction with empirical data in China, *Information Sciences* 179 (1–2) (2009) 89–108.
- [37] L.R. Medsker, *Hybrid Intelligent Systems*, Kluwer Academic Publishers, 1995.
- [38] F. Min, Q. Liu, A hierarchical model for test-cost-sensitive decision system, *Information Sciences* 179 (14) (2009) 2442–2452.
- [39] E. Oja, Neural networks, principal components and subspaces, *International Journal of Neural Systems* 1 (1989) 61–68.
- [40] E. Oja, H. Ogawa, J. Wangviattana, Principal components analysis by homogeneous neural networks, Part 1. The weighted subspace criterion, *IEICE Transaction on Information and Systems* E75D (1992) 366–375.
- [41] C. Romero, *Teoría de la decisión Multicriterio: Conceptos, Técnicas y Aplicaciones*, Alianza Editorial, 1993.
- [42] P.J. Sánchez, L. Martínez, C. García-Martínez, F. Herrera, E. Herrera-Viedma, A fuzzy model to evaluate the suitability of installing an enterprise resource planning system, *Information Sciences* 179 (14) (2009) 2333–2341.
- [43] Z. Sun, G. Finnie, *Intelligent Techniques in E-Commerce: A Case-Based Reasoning Perspective*, Springer, Berlin, Heidelberg, 2004.
- [44] Z. Sun, G. Finnie, MEBRS: a multiagent architecture for an experience based reasoning system, in: R. Khosla, R.J. Howlett, L.C. Jain (Eds.), *Proceedings of the 9th International Conference Knowledge-Based Intelligent Information and Engineering System, KES'05, LNAI*, vol. 3681, Springer, Berlin, 2005, pp. 972–978.
- [45] S. Thomassey, A. Fiordaliso, A hybrid sales forecasting system based on clustering and decision trees, *Decision Support Systems* 42 (1) (2006) 408–421.
- [46] M. Vassileva, Operative planning of the production program in a textile enterprise with the help of MKO-1 software system, *Cybernetics and Information Technologies* 1 (2006) 58–68.
- [47] I. Watson, F. Marir, Case-based reasoning: a review, *The Knowledge Engineering Review* 9 (4) (1994) 355–381.
- [48] I. Watson, *Applying Case-based Reasoning: Techniques for Enterprise Systems*, Morgan Kaufman, San Mateo, CA, 1997.
- [49] M. Wooldridge, N.R. Jennings, *Agent Theories, Architectures, and Languages: A Survey*, Intelligent Agents, Springer-Verlag, 1995.
- [50] M.L. Wong, A flexible knowledge discovery system using genetic programming and logic grammars, *Decision Support Systems* 31 (4) (2001) 405–428.
- [51] J.C. Yañez, *Importancia del sistema de control interno en la auditoría legal, contrastes empíricos*, Ph.D. Thesis, Universidade de Vigo, Spain, 2003.