AGMI
An Agent-Mining Tool and its Application to Brazilian Government Auditing

Carlos Vinícius Sarmento Silva
Controladoria-Geral da União, SAS, Qd 01, Bl A, Edifício Darcy Ribeiro, Brasília/DF, CEP 70.070-905, Brazil
carlos.silva@cgu.gov.br

Célia Ghedini Ralha
Computer Science Department, University of Brasília, POBox 4466, Brasília/DF, CEP 70.904-970, Brazil
ghedini@cic.unb.br

Keywords: Agent-mining tool, Brazilian Government Auditing, AGMI.

Abstract: This paper presents research combining two originally separated areas increasingly interrelated: distributed multi-agent systems and data mining. In our approach, we prove the interaction features in a bilateral and complementary way, since we have defined an integrated architecture and developed a prototype, which has been used in a government auditing study case. In Brazil, government auditing is performed by the Office of the Comptroller General (CGU), where several approaches are being used to prevent and fight corruption. However, some activities such as government purchasing fraud detection are limited by the difficulty in finding effective ways to implement. Considering data mining perspective, we have used different model functions, such as clusterization and link analysis with association rules. Our approach integrating multi-agent and data mining techniques resulted in expressive discovered knowledge, which would help detection of cartels acting in public bidding processes at CGU.

1 INTRODUCTION

The CGU is the agency of the Federal Government in charge of assisting the president of Brazil in matters which, within the executive branch, are related to defending public assets and enhancing management transparency through internal control activities, public audits, corrective and disciplinary measures, corruption prevention and combat, and coordinating ombudsman's activities. As the internal audit unit and the anti-corruption agency of the Brazilian Federal Government, CGU implements the following actions: (i) Internal Control with guidance to public managers and audits, (ii) referral of audit results to the agencies with responsability for enforcement of the applicable sanctions, (iii) disciplinary actions and (iv) corruption prevention (`http://www.cgu.gov.br/english/`).

Nowadays, a large volume of information has been produced and stored by the Brazilian government information systems. Considering only 2009, the Federal Accountability System (SIAFI) registered one billion of financial transactions. All this data are normally used to support the preparation and execution of government auditing. In this way, CGU has driven efforts to apply technologies in order to promote transparency and corruption prevention. However, the analysis of the available data to produce useful knowledge to auditing activities is a hard task. Data mining (DM) and knowledge discovery in large databases are playing an important role, especially when integrated to other computational techniques to Analise and explore information.

In the past decade, agents/multi-agent systems (MAS) and DM/knowledge discovery have emerged as two increasingly interrelated research areas, opening space to the agent-mining interaction and integration (AMII) research field. This new field has driven efforts from both sides to find benefits and complementarity to both communities (Cao, 2009; Ralha, 2009).

In the context of CGU and the Brazilian Government, this paper presents AGMI, an agent-mining tool, which has been used to Brazilian Government Auditing to help prevent corruption through knowledge discover at the Brazilian Federal Bidding database entitled ComprasNet (`http://www.comprasnet.gov.br/`).

The rest of this paper is divided as follows: in Section 2, we present an overview of related areas; in Section 3, we present AGMI architecture and prototype; in Section 4, we discuss the application problem and solution considering the Brazilian Government Auditing; and finally, in Section 5, we conclude and suggest future work.

2 RELATED AREAS OVERVIEW

In this section, we intend to present a brief survey of MAS and DM techniques, considering the existence of AMII area of study. The AMII area benefits from the possibilities offered by the distributed MAS to improve overall DM performance. We consider this content vital as it's not concerned with particular DM techniques, but with the collaborative work of distributed agents in the design of MAS.

2.1 Agent and MAS

Generically, an Intelligent Software Agent (ISA) uses Artificial Intelligence in the pursuit of goals (Luger, 2002; Russell and Norvig, 2009). Thus, an ISA is a software able to do autonomous actions in some environment in order to meet its design objectives (Wooldridge, 2009).

Considering ISA features and properties we might notice the importance of a MAS. In the literature, we find many definitions for a MAS, but mostly they agree as referring to a computational system composed by more than one agent (Huhns and Singh, 1998; Weiss, 2000; d'Inverno and Luck, 2004; Wooldridge, 2009). Thus, a MAS is a system where many agents interact with the environment in a cooperative or competitive way to achieve individual or group objectives.

According to Wooldridge (2009), an agent is a computing entity having four features: autonomy, reactivity, interaction and initiative, and the autonomy is the main characteristic of an agent. In other words, agents are able of acting independently, exhibiting control over their internal state, in opposition to traditional event driven approaches.

Considering the reactivity feature of MAS, Wooldridge (2009) defines a reactive system as one that maintains an ongoing interaction with its environment, and answers to changes that occur in it. Thus, this type of agent just perceives and reacts to environment stimulus, what can be improved with a goal-oriented approach.

In this direction, the pro-activeness feature of agents can be implemented by a goal directed behavior. In this case, an ISA generates and attempts to achieve goals, not driven solely by events, but taking the initiative and recognizing opportunities. These leads to the open problem of balancing reactive and goal-oriented ISA behavior. A good ISA project involves designing adequate balance among these two different kind of agent behaviors. Thus, we want agents to be reactive, answer to changing conditions in an appropriate time, and autonomous, systematically working towards long-term goals by acting in a pro-activeness fashion.

Compared with a client−server centralized system, the advantages of MAS include distribution of processing, support for a more flexible peer-to-peer model, decentralization of control, reduction of network bandwidth use and scalability (Meng et al., 2007).

But MAS is a complex subject, since it involves many different Computer Science areas, like formal methods to represent knowledge and reasoning, language theory, communication and interaction protocol technologies, software architecture including high level or meta-level, semantics and ontological methods, algorithms and complexity theories, planning processes, heuristics and meta-level strategies, among others. All these aspects are related to different approaches of the design of agents society.

2.2 Data Mining Techniques

DM and Knowledge Discovery in Databases (KDD) promise to play an important role in the way people interact with databases, especially scientific databases where analysis and exploration operations are essential (Fayyad, 1997). As cited by the author, our ability to analyze and understand massive datasets lags far behind our ability to gather and store the data.

Fayyad et al. (1996a) cites that a new generation of computational techniques and tools is required to support the extraction of useful knowledge from the rapidly growing volumes of data.

DM involves fitting models to or determining patterns from observed data. The fitted models play the role of inferred knowledge. Deciding whether or not the models reflect useful knowledge is a part of the overall interactive KDD process for which subjective human judgment is usually required. A wide variety and number of DM algorithms are described in the literature. A brief review of specific popular DM algorithms can be found in (Fayyad et al., 1996a; Fayyad et al., 1996b).

According to Fayyad et al. (1996a), the common model functions in current DM practice include:

- Classification: maps (or classifies) a data item into one of several predefined categorical classes.

- Regression: maps a data item to a real-value prediction variable.
- Clustering: maps a data item into one of several categorical classes (or clusters) in which the classes must be determined from the data unlike classification in which the classes are predefined. Clusters are defined by finding natural groupings of data items based on similarity metrics or probability density models.
- Summarization: provides a compact description for a subset of data. A simple example would be the mean and standard deviations for all fields. More sophisticated functions involve summary rules, multivariate visualization techniques, and functional relationships between variables. Summarization functions are often used in interactive exploratory data analysis and automated report generation.
- Dependency modeling: describes significant dependencies among variables. Dependency models exist at two levels: structured and quantitative. The structural level of the model specifies (often in graphical form) which variables are locally dependent; the quantitative level specifies the strengths of the dependencies using some numerical scale.
- Sequence analysis: models sequential patterns (e.g., in data with time dependence, such as time-series analysis). The goal is to model the states of the process generating the sequence or to extract and report deviation and trends over time.
- Link analysis: determines relations between fields in the database (e.g., association rules (Agrawal et al., 1996) to describe which items are commonly purchased with other items in grocery stores). The focus is on deriving multifield correlations satisfying support and confidence thresholds.

According to Han and Kamber (2005), the coverage of an association rule is the number of instances for which it predicts correctly – this is often called its support. Its accuracy often called confidence – is the number of instances that it predicts correctly, expressed as a proportion of all instances to which it applies. For example, with the rule:

```
If temperature=cool then humidity=normal
```

the coverage is the number of days that are both cool and have normal humidity, and the accuracy is the proportion of cool days that have normal humidity (100% in this case).

3 AGMI ARCHITECTURE AND PROTOTYPE

DM has many characteristics which make interaction and integration to MAS possible by combining knowledge discovered through the different DM techniques. In this section, we present our architecture of cooperative DM, which uses two frameworks, JADE and DMA:

- Java Agent Development Framework (JADE) is a Java implemented framework to support software agents implementation. Besides portability being assured by the Java language, the framework allows the implementation of the MAS in a distributed way, which facilitates processing large amounts of data (Bellifemine et al., 2007).
- Data Mining for Agents (DMA) is a framework developed to enable integration of DM algorithms and a multi-agent environment. The framework was based on the Weka, an open source software which is composed by a collection of machine learning algorithms for DM tasks. The algorithms can either be applied directly to a dataset through the program or called from your own Java code (Waikato, 2009).

In our approach, some features have been developed and some algorithms were modified in order to adapt them to multi-agent environment and facilitate the integration of DM activities.

Figure 1 presents AGMI architecture with three layers: user, management and operational. These layers are connected by the JADE Platform, which may be distributed among different physical hosts. The user interface is used to select the attributes and DM techniques, and then to start the knowledge discovery process. After the process ending, the result is presented to user validation.

The management layer focuses on the major activities of control and interaction organization among mining agents. In this layer, the datasets are prepared in accordance with the specification of each DM algorithm. The management layer agents also negotiate the services required, in order to continue the process of knowledge discovery. In this layer, knowledge found is evaluated before being presented to user.

In the operational layer, the mining agents perform specific DM algorithms. In this layer, there may be more than one agent running the same algorithm to attend the service demand of one or more DM techniques.

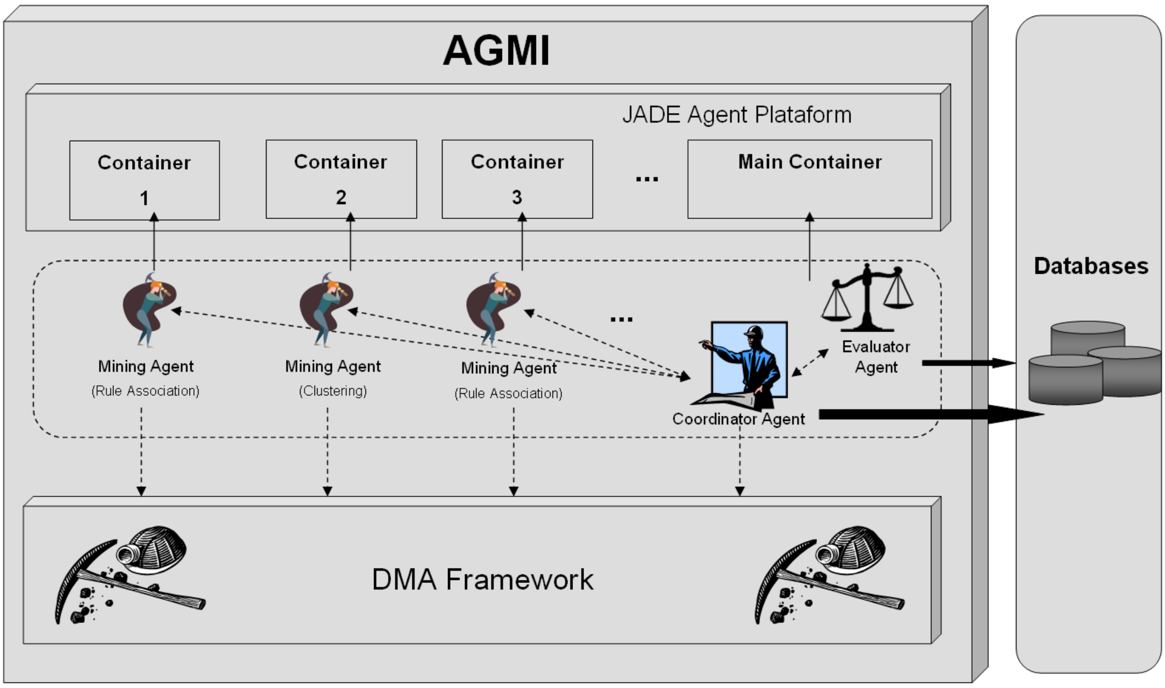

Figure 1: AGMI Architecture Schema.

3.1 Agents

Our model uses four kinds of agents: mining agents, coordinator agent, evaluator agent and interface agent, as described bellow.

Mining agents are responsible for running DM algorithms. The architecture is designed in a way that each agent carries out a specific mining algorithm. Thus, we have multiple agents running the same or different techniques of DM, and using the same or different algorithms. The DM services are classified by DM technique, which can have multiple mining agents providing the same service. Some examples of services are: Service Association Rules, Clustering Service, Attribute Selection Service, among others.

The core of the agent is simple and implements only a specific mining algorithm, having the ability to negotiate its service. This negotiation consists of offering the service and reporting its memory capacity and processing power available to perform the task. An agent only offers its service when it is not already performing another task.

In general, DM tasks require a lot of computer resources in the processing of large volumes of data. Therefore, it may be interesting to distribute mining agents into distinct hosts, but this isn't mandatory. This is possible in our model because of the distributing ability of the JADE Platform.

Coordinator Agent is the agent responsible for overall coordination of the activities of DM. It prepares datasets for the mining agents and coordinates the interactions allowing the cooperation between different techniques of DM.

The coordinator negotiates with mining agents will be hired when the service is needed. For this negotiation, it is necessary to consider the resources of each mining agents to perform the service. If a particular mining service is necessary, and more than one agent is able to execute, the coordinator examines which of them has the best profile to perform the task, considering memory analysis and processing resources of each available agent.

Evaluator Agent is another management layer agent. In general, DM algorithms work with evaluating criteria. For instance, support and confidence criteria are used to find and sort association rules (Section 2.2). However, it is possible to adopt other criteria to sort discovered knowledge, and it can be done by the Evaluator Agent. We can adapt this agent to perform a specific function or just define the minimal values of covering and accuracy of rules resulting from mining algorithms.

Interface Agent helps the user to set the main parameters of the knowledge discovery process such as database and attributes selection, among others. When the process finishes, this agent reports the discovered knowledge to the user for validation.

3.2 Communication and Interaction Protocol

The communication in our model is necessary for cooperation among agents. Moreover, agents are autonomous entities, and communication is the instrument that brings the synchrony between the agents to reach the ultimate goal, which is the knowledge discovery.

JADE platform provides the implementation of FIPA ACL, a message based protocol defined by Foundation for Intelligent Physical Agents (FIPA), an institution that sets standards for development of technologies based on intelligent agents (http://www.fipa.org/).

FIPA specifies a set of standard interaction protocols, which can be used as standard templates to build agent conversations. One of these interaction protocols is the FIPA Contract Net protocol. According to (FIPA, 2002), that protocol is a minor modification of the original Contract Net Interaction Protocol (IP) pattern, in which rejection and confirmation communicative acts are added.

In the Contract Net IP, one agent (the Initiator) takes the role of manager, which wishes to have some task performed by one or more agents (the Participants); and further, wishes to optimize a function that characterizes the task. This characteristic is commonly expressed as the price, in some domain specific way, but could also be soonest time to completion, fair distribution of tasks, among others. For a given task, any number of Participants may respond with a proposal; the rest must refuse. Negotiations then continue with the Participants that proposed.

JADE provides an implementation of the FIPA Contract Net protocol (Bellifemine et al., 2008). In AGMI architecture, the negotiation is established among the Coordinator Agent and the Mining Agents. The Coordinator announces the task to be performed by Mining Agents and waits for the proposals sent back.

In this architecture, the expected proposals are the available resources for the agents. A specific DM task can need more available memory than another one, for instance. The Coordinator then manages the Mining agents according to their available resources, based on an internal knowledge base that associates the DM tasks with the optimal agent configuration. Agent services can be found at Yellow Pages, in JADE. It is a service provided by JADE Platform that publishes the registered agents and their services.

4 APPLICATION PROBLEM AND SOLUTION

In general, the identification of cartels is a difficult task since it requires analysis of several public bidding processes, which usually exceeds the scope of only one government department. Cartels can operate in various government departments, cities and even states of the Federation.

The analysis of data from databases using languages, such as Structured Query Language (SQL) queries, is also impractical because of the exponential solution space. Thus, given the difficulties inherent in the process of detecting cartels, auditing activities involving this issue are limited only to confirm suspicions raised normally after denunciation.

So there is no deterministic way to identify cartels effectively, because the solution space is exponential in the number of companies participating in bidding processes analyzed. Thus, the problem consists in creating an efficient way to identify groups of companies, which might be suspected of practicing cartels in public bidding processes.

4.1 A Solution using Association Rules

In Silva and Ralha (2010), we propose a solution using Association Rules to solve the problem of cartels detection in public bidding processes. The proposal is due to the fact that this technique is useful to find strong relationships between attributes. So, it is possible to aply this technique creating a dataset, so each attribute is boolean and represent a company of the database. What confirms its participation or not in a bidding process. Then, we fill that dataset with the bidding processes from database, defining which company has participated ('true' value) or not ('false' value) of each bidding process.

The dataset for association rule technique must be constructed as the matrix A consisting of m rows and n columns:

$m =$ (total of bidding processes from database)
$n =$ (total of companies from database)

$$a_{i,j} = \begin{cases} true & \text{if company } j \text{ has participated} \\ & \text{of the bidding } i; \\ false & \text{if company } j \text{ has not participated} \\ & \text{of the bidding } i; \end{cases}$$

$$1 \leq i \leq m; 1 \leq j \leq n;$$

Thus, we expect to obtain rules like the following:

$$company_A = true,\ company_B = true \rightarrow company_C = true$$

To obtain such kind of rule, we have to suppress rules with attributes indicating the non-participation of companies, because we are interested in finding cartels. What means participation in bidding processes of all companies in a specific group.

4.2 Solution Using Multi-Agent Architecture

The database used in the experiment was based on a Brazilian Federal Bidding System – ComprasNet, where the bidding processes of Federal Government are conducted online. These data are relative to all bidding processes of a specific type of service contracted by Federal Executive agencies between the years 2005 and 2008 in all states of the Federation.

Table 1 shows information about the dataset used in our experiments. Each record in the dataset represents one bid of a company in a specific bidding process.

Table 1: Database of the Experiment

Information	Number
Records	26615
Bidding Processes	2701
Companies	3051
Companies with at least 1 victory	1162
Companies with at least 5 victories	121

4.3 First Experiment

To validate our architecture, we started running a test with three agents: coordinator, evaluator and rule association agent. The last one was essential to solve our problem of cartels and will be described.

Rule Association Agent is responsible for implementing the algorithm of association rules. This technique can be used to solve the problem of cartels, as we presented earlier (Silva and Ralha, 2010). This agent runs the *Apriori* algorithm, available in Weka framework, which has been adapted to DMA in order to be used in a multi-agent environment. In previous tests, we found that setting high values in the configuration of minimal support for execution of this algorithm does not guarantee good rules to identify cartels. A rule with high lower bound of support might just imply the presence of big companies in bidding processes.

Thus, setting a high lower bound of support in this algorithm can suppress the appearance of several good rules, with real features of cartels. By the other hand, high lower bound of confidence ensures the selection of good rules. The Association Rule Agent configuration is shown in Table 2.

Table 2: Association Rule Agent Settings

Lower Bound Support	$10 * \frac{100}{inst.number}$
Lower Bound Confidence	90%

We have noticed the more we decrement the support minimal value, more rules are returned. Thus, we have to sort the best rules to Analise. Furthermore, to discover cartels, we have to consider how many contracts the suspicious group has won in public biddings.

With the help of experts, we have defined a method for evaluating the rules obtained through the process of DM. The evaluation formula has been defined as presented in Equation 1.

$$RQ = 100.\frac{V(C)}{Sup. \times Inst.} \qquad (1)$$

$$\begin{aligned}
RQ &= \text{rule quality;}\\
Sup. &= \text{rule support value;}\\
Inst. &= \text{total number of instances;}\\
C &= \text{company set from rule;}\\
B &= \text{bidding processes where the company group } C \text{ has participated;}\\
V(C) &= \text{number of victories in } B \text{ of any company in } C.
\end{aligned}$$

This function was implemented in the Evaluator Agent for measuring the quality of the rules, and means the probability of one company of the suspicious group winning a bidding process. The higher quality value, the more suspicious the group is of cartel practicing.

Table 3: Agents distribution

Host A	Intel Core 2 2.40 GHz, 2.00 GB RAM	
Host B	Intel Pentium Dual 1.86 GHz, 2.00 GB RAM	
	Host A	**Host B**
Coordinator Agent	X	
Evaluator Agent	X	
Assoc. Rules Agent		X

The agents were distributed as shown in Table 3. The test results is shown in Table 4.

Figure 2 shows the numbers of the top ten rules discovered. The best rule, according to Equation 1 has

Table 4: Results of Experiment 1

Information	Number
Rules found	128
Execution time	29'
Average RQ (Top 100)	16.56
Average Support (Top 100)	30.98
Top 10 rules	
Average RQ	33.00
Average Support	26.90

46 points of quality and 26 of support value. We can notice in Table 4 the difference between the average support of all results and the top ten rules. The top ten rules have a smaller average support. Thus, we note that higher limits of support do not guarantee better rules.

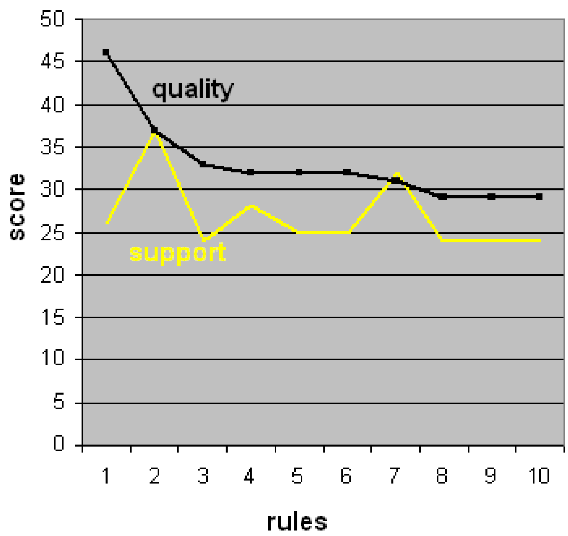

Figure 2: Top ten rules of the First Experiment.

4.4 Second Experiment

In the First Experiment, with only one mining agent in the system (Rule Association), we couldn't set the lower bound support to less than 24. It happened due to the lack of memory resource available in our machine. The more data to process and lower the bound support set, the more memory is consumed by the rule association algorithm.

Furthermore, it's quite possible to have rules with support less then 24 with real characteristics of a cartel. On the other hand, cartels usually act in specif regions. Thus, we needed to introduce a strategy to divide our space to seek the groups more accurately. For this, we introduced a clustering mining agent to find regions where companies join to participate to bidding processes.

Clustering Agent implements the Expectation-Maximization (EM) algorithm to discovery clusters considering the companies and the states, where they have participated of a bidding processes. According to Han and Kamber (2005), the EM algorithm extends the k-means paradigm in a different way. Whereas the k-means algorithm assigns each object to a cluster, in EM each object is assigned to each cluster according to a weight representing its probability of membership. In other words, there are no strict boundaries between clusters.

Therefore, new means are computed based on weighted measures. That characteristic is very important to our problem, considering that one cartel can also act in more than one region, or in this case, one cluster. The algorithm is available in Weka and was added to DMA Framework of AGMI. In our system, the Clustering Agent was prepared to process data relative to two attributes: the company and the state, where it has participated in public biddings. For this experiment, the Clustering Agent was added in the Host A (Table 3).

4.4.1 Agent Interaction

Figure 3 presents the interaction among our agents. Before the Coordinator start the tasks, all mining agents have to register their services in the JADE Yellow Pages (Bellifemine et al., 2008). It is quite possible to have more than one agent providing the same service, then the Coordinator chooses based on the Contract Net protocol and other designed criteria.

To solve our problem, the system was designed to start running the rule association service on the entire database, like in the First Experiment (Section 4.3). The Coordinator delegates this service after the negotiation, but now at the same time, it is also ordered to execute the clustering service on the database (2).

As soon as the result of the clustering agent is returned, all clusters found are adjusted to generate one dataset relative to that specific cluster of states found, in order to define geographic regions to look for cartels evidences. Thus, the Coordinator generates the datasets and then it delegates rule association services in all datasets defined by the clusters.

The results from Rule Association Agents are sent by the Coordinator to the Evaluator Agent (4). These agents will filter and sort the best rules to finally present to user.

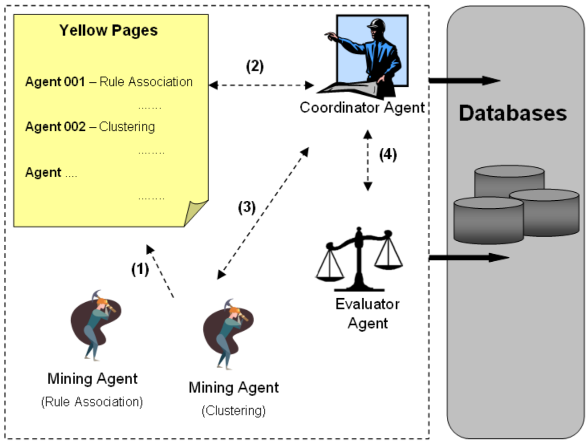

Figure 3: Interaction among agents.

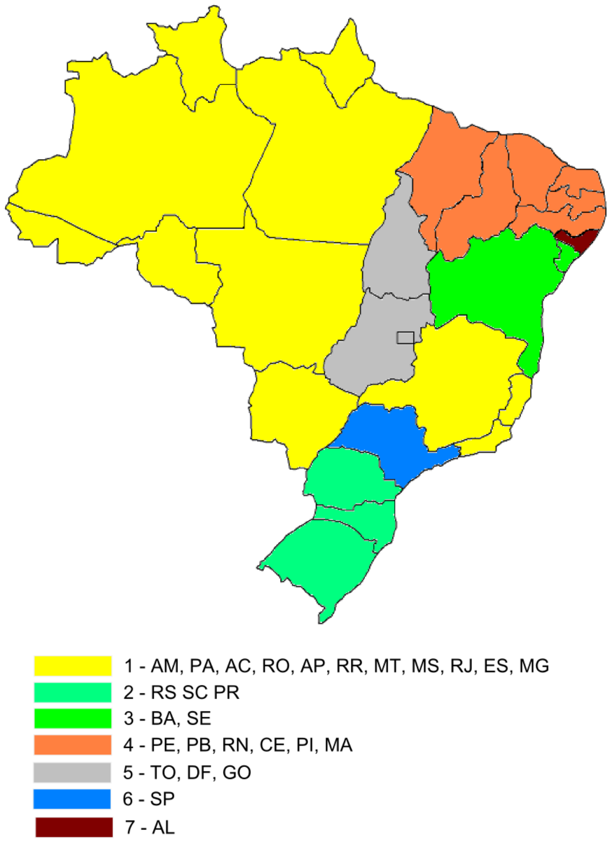

Figure 4: Clusters found by the Clustering Agent.

4.4.2 Results

Our main goal is to find association rules that indicate suspicious groups of cartel acting. However, with the addiction of the Clustering Agent, we found the regions of public biddings, and the companies that participate of those biddings in each region. Seven clusters were found and they can be seen in Figure 5.

We expected that all clusters were made up by states with common borders. In general, the companies do not act in states far from each other. However, the results have shown one cluster with large proportions and with some states very disconnected.

The test results is shown in the Table 4. We can notice that the addition of the Clustering Agent improved the results dramatically. Considering the top ten rules of both experiments, the values for quality of rules has grown up more than 150%. If we compare the average of the Top 100 rules, the Second Experiment presents an average more than 4 times greater than the First Experiment average.

With the addition of the Clustering Agent in our tests, the search space was divided, and so, it was possible to reduce the lower bound of support generating more and best rules. Thus, we improved a lot the quality of rules produced by the system. This proves the potential to integrate different DM techniques to SMA.

Table 5: Results of Experiment 2

Information	Number
Rules found	6150
Execution time	75’
Average RQ (Top 100)	69.75
Average Support (Top 100)	10.50
Top 10 rules	
Average RQ	89.70
Average Support	9.40

4.5 Discovered Knowledge

The experiments present lots of rules, specially the second one (Section 4.4), due to the value of lower bound of support. Thus, we had to implement a filter in the Evaluator Agent to choose the best rules and sort them. So, the rules presented by the Evaluator Agent after the execution showed lots of company groups with evidence of cartel acting.

The Clustering Agent was originally introduced in the system just for dividing the solution space in order to facilitate the data processing. However, the discovered clusters revealed the trends of companies participation in public bidding in Brazil. For example, Cluster 1 showed a trend of companies from Rio

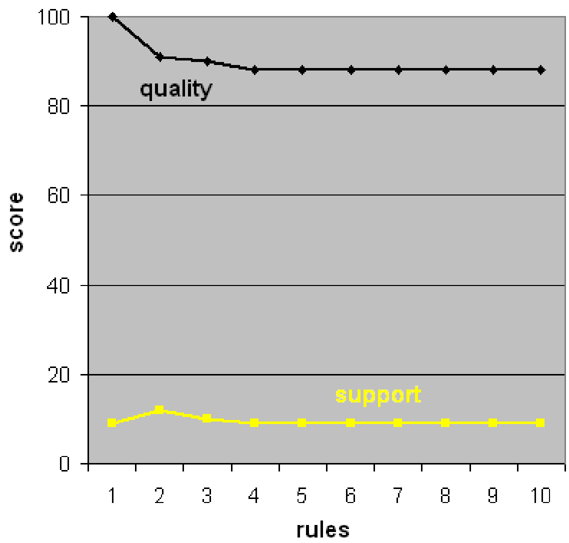

Figure 5: The Top ten rules of the Second Experiment.

de Janeiro to explore public bidding market of Minas Gerais and other states of north region. Another group had participated of public biddings in three different states of Southeast region, two states of Center-West region and one of North region, probably pushing the Cluster 1 formation.

We present some of the best rules as example of discovered knowledge:

- In 9 different public biddings in the same state, and in just one government agency, one rule has pointed the participation of two specific companies. In spite of the fact that both companies have participated of all biddings, just one of the companies has won all of the biddings. When we analyzed the history of the loser company, we found that it had only participated in those 9 exactly bidding processes, evidencing a possible simulation of competition to hide the cartel characteristics. Probably the company was created just for creating an artificial competition in the public biddings, where competition is mandatory.
- A group with cartels characteristics, made up by 4 companies, has acted in Region 2 in 10 different public biddings, and has won 9 of these. Two companies of that group have appeared in other 7 different groups, in the same region, discovered by other rule association.
- Three companies were detected in a rule with 14 public biddings participation in common. They won 8 of those biddings. When we checked their log of public bidding participation we found an average of around 30 participations for each one. A relatively low average of participation, considering 3 years of log.

5 CONCLUSIONS

Apart from being in charge of inspecting and detecting frauds in the use of federal public funds, the CGU is also responsible for developing mechanisms to prevent corruption. The idea is that, besides detecting cases of corruption, CGU has the role of acting proactively by developing means to prevent their occurrence. Thus, in this article we presented the Agent-Mining Tool – AGMI, a MAS based tool to support distributed DM activities for the Brazilian Government Audition context.

AGMI includes the DMA – a Weka-based framework – made up by a set of DM algorithms adapted to multi-agent environment in order to support the mining agents layer. AGMI architecture has also a management layer, which enables the combination of distinct DM techniques, in order to improve the knowledge discovery process.

We tested AGMI as a proposed solution to the problem of detecting cartels in public biddings. Two experiments were run and the results obtained showed the AGMI potential as solution for this problem. Several association rules indicating evidences of cartel acting in public biddings, as well as some rules suggesting fraudulent simulation of competition in federal biddings, were found.

The first experiment was executed with only one mining agent (rule association) in the operational layer, whereas in the second one, we added a Clustering Agent. The results of the second experiment showed that the average quality of the top ten association rules increased around 150%. If we compare the average quality of the top 100 rules, the second experiment presented an average more than four times greater than the first one.

Besides, the clusters found by Clustering Agent showed the regional action trends of companies in the public bidding market. In some regions, the found clusters also helped to detect cartel behaviors.

In future work, we will study other ways to automatize the attribute selection, and reduce the processing time. For that, we are studying other mechanisms in order to improve our agent interaction protocol. Besides, aiming to enrich the knowledge discovery process, some heuristics will be studied to be applied in our model, and genetic algorithm will be tried on, as well.

REFERENCES

Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., and Verkamo, A. I. (1996). Fast discovery of association rules. pages 307–328.

Bellifemine, F., Caire, G., Trucco, T., and Rimassa, G. (2008). Jade programmers guide.

Bellifemine, F. L., Caire, G., and Greenwood, D. (2007). *Developing Multi-Agent Systems with JADE*. Wiley.

Cao, L. (2009). Introduction to agent mining interaction and integration. In Cao, L., editor, *Data Mining and Multi-agent Integration*. Springer US.

d'Inverno, M. and Luck, M. (2004). *Understanding Agent Systems*. Springer Series in Agent Technology, Berlin, Germany, 2^{nd} revised and extended edition. ISBN 3-540-40700-6.

Fayyad, U. M. (1997). Data mining and knowledge discovery in databases: Implications for scientific databases. In *SSDBM '97: Proceedings of the Ninth International Conference on Scientific and Statistical Database Management*, pages 2–11, Washington, DC, USA. IEEE Computer Society.

Fayyad, U. M., Piatetsky-Shapiro, G., and Smyth, P. (1996a). From data mining to knowledge discovery: an overview. pages 1–34.

Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R., editors (1996b). *Advances in knowledge discovery and data mining*. American Association for Artificial Intelligence, Menlo Park, CA, USA.

FIPA, F. F. I. P. A. (2002). Fipa contract net interaction protocol specification. http://www.fipa.org/specs/fipa00029/SC00029H.html.

Huhns, M. N. and Singh, M. P., editors (1998). *Readings in Agents*. Morgan Kaufmann, San Francico, USA. ISBN 1-55860-495-2.

Luger, G. F. (2002). *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*. Addison-Wesley, USA, 4th edition. ISBN 0-201-64866-0.

Meng, A., Ye, L., Roy, D., and Padilla, P. (2007). Genetic algorithm based multi-agent system applied to test generation. *Comput. Educ.*, 49(4):1205–1223.

Ralha, C. G. (2009). Towards the integration of multiagent applications and data mining. In Cao, L., editor, *Data Mining and Multi-agent Integration*. Springer US.

Russell, S. and Norvig, P. (2009). *Artificial Intelligence: A Modern Approach*. Prentice Hall Press, Upper Saddle River, NJ, USA.

Silva, C. V. S. and Ralha, C. G. (2010). Utilizao de tcnicas de minerao de dados como auxlio na deteco de cartis em licitaes. *XXX Congresso da Sociedade Brasileira de Computao*.

Waikato, U. (2009). Weka machine learning project. http://www.cs.waikato.ac.nz/ ml/index.html.

Weiss, G. (2000). *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. The MIT Press. ISBN 0-262-73131-2.

Wooldridge, M. (2009). *Introduction to MultiAgent Systems*. John Wiley & Sons, 2nd edition. ISBN 978-0-470-51946-2.