Combining attack trees and fuzzy numbers in a multi-agent approach to fraud detection

Alessandro Buoni

IAMSR, Turku Centre for Computer Science, Joukahaisenkatu 3-5 B, 20520 Turku, Finland E-mail: abuoni@abo.fi.

Mario Fedrizzi

Department of Computer and Management Sciences, University of Trento, Via Inama 5, 38122 Trento, Italy,
Email:mario.fedrizzi@unitn.it.

József Mezei

IAMSR, Turku Centre for Computer Science, Joukahaisenkatu 3-5 B, 20520 Turku, Finland E-mail: jmezei@abo.fi.

Abstract: The fraud surveys carried out in the last five years by leading international consulting companies demonstrate that fraud is an increasing phenomenon depending most of all on behavioral aspects. Therefore, when addressing fraud detection processes the adoption of traditional statistical techniques comes out to be not as adequate as those based on the evaluations of experts working in a multi-agent framework. In this paper we introduce a multi-agent system called Fraud Interactive Decision Expert System (FIDES), which puts more emphasis on the evaluation of behavioral aspects of fraud detection according to the judgments expressed by two groups of experts, inspectors and auditors respectively. FIDES combines think-maps, attack trees and fuzzy numbers under a Delphi-based team work support system and offers to the users a suitable way to better understand and manage fraud schemes.

Keywords: Fraud detection, Think-maps, Attack trees, Delphi method, Fuzzy numbers.

Biographical notes: A.Buoni is a Phd student at IAMSR, Åbo Akademi University, Finland. His research interests are in fraud detection, multi-agent systems and knowledge management.

M.Fedrizzi is a Professor at Department of Computer and Management Sciences, University of Trento, Italy. His research interests are in decision modelling under uncertainty, fuzzy logic and soft computing.

J.Mezei is a Phd student at IAMSR, Åbo Akademi University, Finland. His research interests are in fuzzy logic and optimization.

1. Introduction

David J. Hand (Hand 2007) points out how institutions in persecuting fraud follow the economic imperative, meaning it does not worth spending $200m to stop $20m fraud. Participants in his study estimate that U.S. organizations lose 5% of their annual revenues to fraud. This means, applied to 2006 U.S. GDP, approximately $652 billion in fraud losses. According to KPMG (KPMG Forensic fraud Barometer 2009), there is a prominent increase of fraud by individuals. Company managers, employees and customers together have been responsible for £300m of fraud in 2008, three times the value of year 2007.

In KPMG Fraud Survey (2009) is shown that the most effective countermeasures for fraud are those ones developed by internal audit using clues given by employee whistleblowers as shown in Fig.1. The survey has been conducted on executives of U.S companies who answered the following question: Through which source do you believe your organization would be most likely to uncover fraud or misconduct?

The embarrassment of admitting to mainly follow an economic imperative in persecuting fraud is coherent with the choice of preferring internal resources on external ones, but this is not only the main reason. It is also related to the awareness that an internal audit team has a better knowledge of their organization, the weakness of internal procedures and the personnel.

Internal Audit	**47 %**
Employee whistleblowers	**20 %**
Line managers	**13 %**
External Auditors	**9 %**
Customers or suppliers	**4 %**
Government regulators or law enforcement	**3 %**
Other means	**2 %**

Fig.1 Fraud countermeasures (KPMG Forensic fraud barometer, 2009)

According to the Advanced Measurement Approaches (AMA), introduced in Basel II accord (Basel Committee 2006), banks are encouraged to develop sophisticated methodologies to calculate the operational risk, monitor the bank activities, reinforce internal control structure and auditing to preserve the integrity of the managerial processes. These systems include also the use of internal and external data, scenario analysis and control factors and an accurate reporting system based on key risk indicators.

In the banking sector there is a prevalence of human fraud. One of the causes encouraging internal fraud is conflict of interests, which limits the effectiveness of the control procedures.

Commonly, the biggest problem of an audit team is to interpret and summarize all these behavioral aspects in order to come up with effective solutions to prevent fraud before they happen or to detect quickly the type of fraud when perpetrated. To this end, audit teams collect information about past behaviors in order to provide a formal representation of the most common typologies of fraud. This way, a repository of domain expert represented by standardized fraudulent attacks can be created and reused for, e.g., playing "what-if" games with potential countermeasures or identifying the nature of new attacks.

Auditors, who are the fraud experts, can use their experience to remove false alarms, but also to detect those crimes which cannot be detected electronically because they are the results of untraceable human behaviors most often perpetrated inside the departments of the bank. The evaluation process of auditors in fraud detection has been examined, e. g., by Bazerman et al. (2007) and by Grazioli et al. (2006), exploiting the reasons of their success and failure, and studying the impact of ambiguity, analyzing a quite extended sample of case studies.

Several authors have demonstrated that a multi-agent approach is particularly suitable to address fraud detection when behavioral aspects play a key role, see for instance Chou et al. (2007), Wang et al. (2009), and Zhang et al. (2008).

We believe that the multi-agent system we are going to introduce in this paper, combining think-maps, attack trees, and fuzzy numbers under a Delphi-based team work support system, do offer to the agents involved (inspectors and auditors) an innovative and suitable way to better understand and manage fraud schemes.

The multi-agent architecture of the Fraud Interactive Decision Expert System (FIDES) (Buoni 2010) will permit them to open up and share their knowledge and then link all the clues in a coherent scheme. The learning by doing approach of think-maps is a good means for inspectors to formally represent their knowledge linguistically expressed, to reconsider their opinions and correct their statements in the light of the comments received by their colleagues.

The paper has been organized as follows. In the second section we introduce the main components of FIDES, i.e. think-maps, attack trees and Delphi method. In the third section we describe the fuzzy mechanism that is used for representing and aggregating the linguistic information provided by the experts of the audit team when addressing the design of the attack tree. In the fourth section the whole structure of the system is figured out combined with the description of a work session. In the last section some conclusions are drawn and future lines of research are sketched out.

2. Think-maps, attack trees, Delphi method

Scope of this paragraph is to provide a synthetic description of the main components of FIDES, focusing the description on those features characterizing the collection and representation of knowledge involved in the analysis and detection of fraudsters' attacks.

2.1. Think-maps

The concept of mind mapping was introduced for first by Tony Buzan in 1974 (Buzan 1974). The idea was to organize keywords into a radiant structure that looks like a tree seen from above (Åhlberg 2007)). A radiant structure permits to put in the middle of the paper the central idea (goal) and through a brainstorming process add around it the branches of the map. Novak and Cañas developed the idea of concept maps (see Novak et.al. 2006), which they defined as "graphical tools for organizing and representing knowledge".

Oxman introduces the idea of think-maps (Oxman 2004), which means that conceptual mapping of designing ideas can be constructed in larger structures where it is possible to organize knowledge acquired by the learner and make it explicit. The software they develop to create these maps is called "Web-Pad". The theoretical basis of think-maps is constructivism. Constructivist theories of learning state that the learner is not a passive recipient of knowledge, but it has an active role in creating knowledge, based on the "learning by doing" approach.

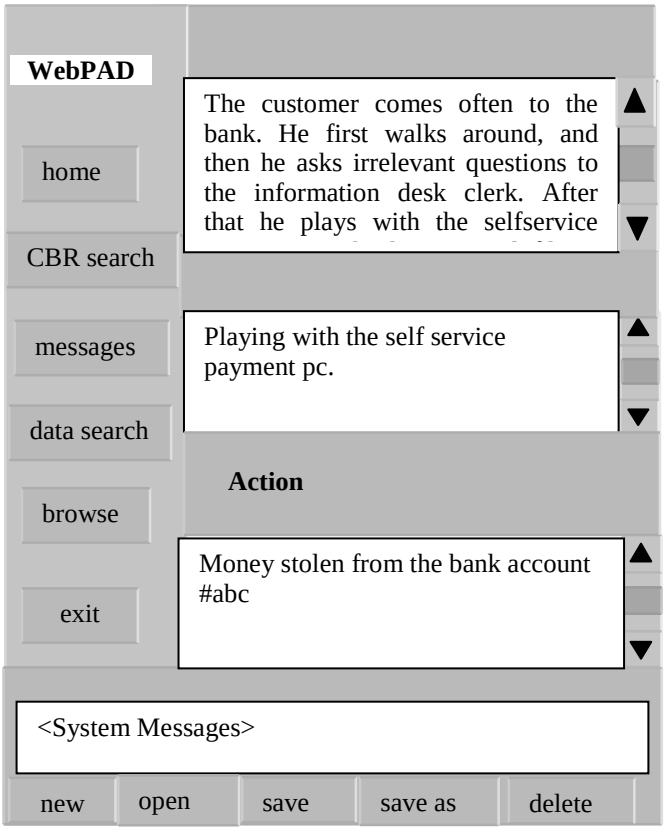

Figure 2. Web-Pad interface (Oxman 2004)

Learners construct their knowledge based on their experience and relationship with concepts. Think-maps have a formal representation called ICF (Issue-Concept-Form), which acts as "a structuring ontology for the construction of conceptual networks of design concepts"(Oxman 2004).

Internal fraud cases can be anticipated by particularly suspicious behaviours like working overtime, reluctance to take breaks and sudden change of lifestyle. In this paragraph will we describe a case to show the process and how inspectors would model it.

In our case inspectors will use Web-Pad (Fig.2) to comment suspicious behaviours and/or activities they observed or information acquired by whistleblowers or insert their reports in order to share them with all the other colleagues. Since the software works in a real time discussion environment their comments will appear in the text box named "Description" as a common chat discussion shape. At this stage inspectors will have the possibility to arrange the text in order to have a clear description of the case.

At this point inspectors will start to underline keywords as it can be seen in the following example:

The customer comes often to the bank. He first walks around, then he asks irrelevant questions to the information desk clerk. After that he plays with the self-service payment pc, checking around if there is someone observing him.

Then keywords, which are considered important to interpret the case, will be associated to 2 main labels "suspicious behaviours" and "action", which are related to one or more suspected.

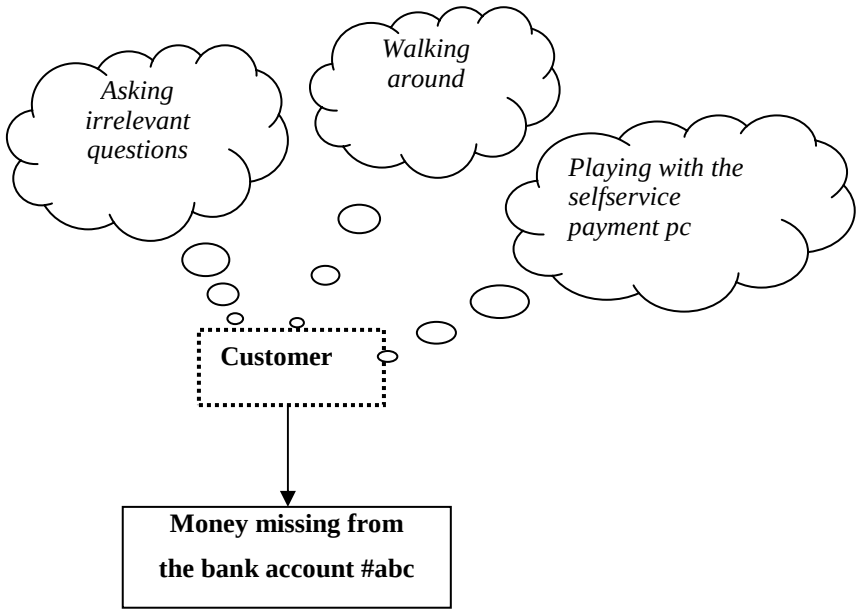

Figure 3. The think map

At the end of this process the think-map shown in Fig.3 is obtained. In the map we can see the main labels, which have been used to visualize a fraud operation. The cloud callouts represent the activities related to "suspicious behaviour", whose label was previously created with Web-Pad, associated to three different suspected persons. There is a customer who has the habit to play with the self-service payment pc of the bank. In the middle of the figure we have the suspected and on the bottom the fraudulent action, which is the amount of money missing in a specific bank account.

Having this think-map as a descriptive model, inspectors will start to create nodes, which will be sent to the auditors to be connected through the Delphi process to build the attack tree. Web-Pad basically works on two levels. A graphic level where think-maps can be visualized and a text level where different operations can be stored and retrieved for the future, supporting the inspectors when they have to build the think-map.

For instance inspectors could be interested to have a list of the cases associated to a particular behaviour and so on. In the data retrieval mode the system can bring up precedents that inspectors consider similar, according to their subjective judgement.

Users can express the level of similarity between two different cases as a number between 0 and 1. Once the database is populated with a significant number of cases, it will be possible to retrieve and visualize them in descendant order, according to their level of similarity, ready to be examined.

2.2. Attack tree

The attack tree, introduced by Schneier (Schneier 1999) is a tree-based diagram to "systematically categorize the different ways in which a system can be attacked".Nodes are the elementary attacks and the root node is the goal of the attack. Children of a node are refinements of this goal, while leaves are attacks which cannot be further on refined. The process of creating an attack tree starts by identifying the possible attack goals, where each goal forms a separate tree, but there is also the possibility that different trees share nodes and subtrees.

Accordingly, modeling an attack tree is a matter of associating a logical AND and a logical OR with each node, and therefore encouraging a structured representation of events and of the ways they are connected.

This supports the discovering of the most likely avenues of approach for an attack making easier the deployment of the most effective countermeasures. Even though the Schneier's attack trees (illustrated in Fig.4) have been considered from their first appearance a convenient tool to systematically categorize the different modes in which an attack can be carried out, nonetheless their network structure has been criticized for its simplicity e for the lack of well sounded theoretical foundations.

Mauw and Oostdijk (2005) arguing that Schneier's approach to attack trees is semantically not well sounded, provide a generalization based on the observation that an attack tree describes an attack suite and that a node can be connected to a multi-set of nodes (bundle) and may contain several bundles.

But, since our paper is more focused on the way on which the team of experts carry out, in a Delphi-based context, the consensual construction of the tree, than on the complexity of its structure, the nature of the tree is irrelevant and therefore we will adopt the Schneier's approach.

Niitsoo (2010) also points out how it is important to develop attack tree models that take into consideration not only whether the attack is possible or not, which can not tell so much about the likelihood of the attack, but also incentives and possibilities available to the fraudster in order to try to analyze his/her behaviour.

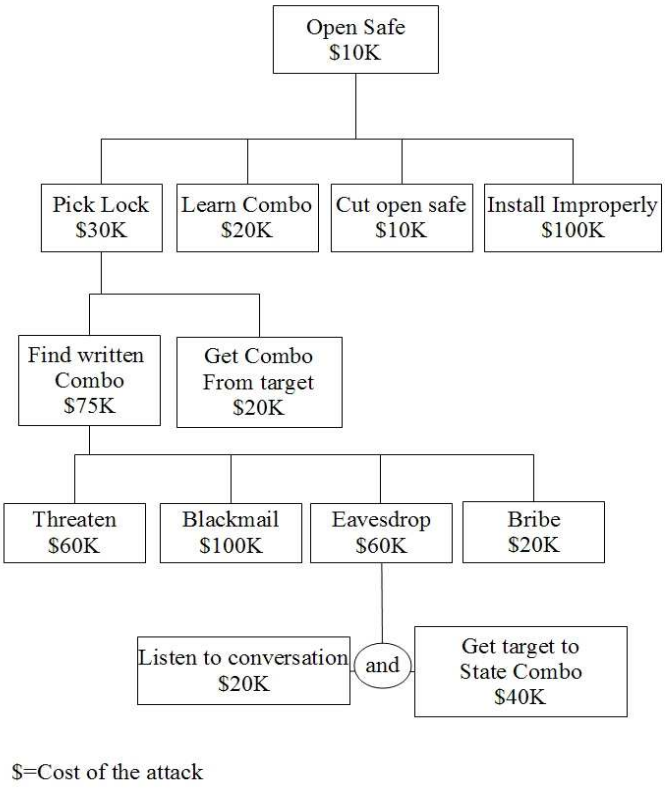

Figure 4. An example of attack tree (Schneier 1999)

2.3. Delphi method

Delphi method, introduced for the first time in the 50s for a U.S. sponsored military project (Gordon 1994), is a systematic, interactive and iterative method which relies on a team of experts, aiming at discussing and structuring the solution of a given problem. The experts are asked to answer questionnaires in two or more rounds, and after each round a moderator provides an anonymous summary of the experts' analysis from the previous round as well as some explanations of their judgments. The moderator encourages

experts to reuse their earlier opinions in light of the outcomes of the analysis provided by the other experts of the team. The process is stopped according to a pre-defined criterion and some average measure of the outcomes of the final round determine the output of the process.

Delphi method has been used also recently in many different field of research like R&D to explore the barrier factors to the adoption of mobile service (Steinert, 2009), security evaluation of embedded systems (Zhang et al. 2008), road safety (Maa et al. 2011) and clinical nursing (McElhinney 2010).

Rowe and Right (1999) suggest four key features for a good design of Delphi:

1. Anonymity of Delphi experts: allows the experts to freely express their opinions without undue social pressures to conform from others in the team. Decisions are evaluated on their merit, rather than who has proposed the idea.

2. Iteration: allows the experts to refine their views in light of the progress of the team's work from round to round.

3. Controlled feedback: informs the experts of the other experts' perspectives, and provides the opportunity for Delphi experts to clarify or change their views.

4. Statistical aggregation of team response: allows for a quantitative analysis and interpretation of output information.

In our system, Delphi will be used as a method for finding the agreement on the connections between different nodes figured out by the inspectors. The role of the moderator will be to ask the experts their opinion about strength of the links connecting different nodes.

The aggregation process will be described in details in section 3. In the end of the Delphi process, the output will be an attack tree.

3. The fuzzy mechanism

The audit team performs the Delphi process aiming to select the nodes and connect them in order to design the attack tree. In the first phase the inspectors determine the possible nodes of the attack tree with the help of the think-map. Then the moderator will ask the experts about the possible connection of the nodes, and aggregate the results to obtain the attack tree. In this paragraph we will describe the fuzzy mechanism which helps the experts to form the attack tree. In the literature, a number of different fuzzy approaches to the analysis of negotiation processes in multi-agent decision making have been proposed, and for an extended overview the interested reader could see, e.g., (Carlsson et al.2004) and (Fedrizzi et al. 1997). Before the description of the model we need some basic definitions from fuzzy set theory.

Definition 1. Let X = x denote a collection of objects (points) denoted generically by x. Then a fuzzy set A in X is a set of ordered pairs

$$A = (x, \mu_A(x)), x \in X \qquad (1)$$

where $\mu_A(x)$ is termed the grade of membership of x in A, and $\mu_A : X \rightarrow M$ is

a function from X to a space M called the membership space. When M contains only two points, 0 and 1, A is non fuzzy and its membership function becomes identical with the characteristic function of a crisp set. This means that crisp sets belong to fuzzy sets. A fuzzy number is a convex fuzzy set on the real line such that

1. $\exists x_0 \in$ A, $\mu_A(x_0) = 1$,
2. μ_A is piecewise continuous.

(The convexity means that all the γ-level sets are convex. Furthermore, we call F the family of all fuzzy numbers).

A γ-level set of a fuzzy set A is defined by $[A]^\gamma = \{x \in$ A: $\mu_A(x) \geq \gamma\}$ if $\gamma > 0$ and $[A]^\gamma = cl\{x \in$ A: $\mu_A(x) > \gamma\}$ (the closure of the support of A) if $\gamma = 0$. Let A be a fuzzy number. Then $[A]^\gamma$ is a closed convex subset of R for all $\gamma \in [0,1]$.

We use the notations

$$a_1(\gamma) = \min[A]^\gamma, a_2(\gamma) = \max[A]^\gamma$$

for the left-hand side and right-hand side of the γ-cut, respectively. When we calculate the arithmetic operations on fuzzy sets (fuzzy numbers), we apply the rules of interval arithmetic. Let A and B be fuzzy numbers with the corresponding γ -cuts: $[A]^\gamma = [a_1(\gamma), a_2(\gamma)]$, $[B]^\gamma = [b_1(\gamma), b_2(\gamma)]$ then the γ-cut of the fuzzy number A + B is the following:

$$[A+B]^\gamma = [a_1(\gamma) + b_1(\gamma), a_2(\gamma) + b_2(\gamma)],$$

and the γ-cut of the fuzzy number αA, where $\alpha > 0$: $[\alpha A]^\gamma = [\alpha a_1(\gamma), \alpha a_2(\gamma)]$.

We will use linguistic labels in the questionnaire and we represent the labels as fuzzy numbers (see Fig.5 for a possible representation). We suppose that the moderator can choose which nodes are parents (V) (with descendant) and which ones are leaves (L) (without descendants, basic attack components). We obtain two sets:

$$L = \{l_1, ..., l_s\}, V = \{v_1, ..., v_t\}.$$

In the first questionnaire the experts have to express their opinion in linguistic terms about statements like " $l_i \in L$ is required for $v_j \in V$ ", for every $i = 1, ..., s, j = 1, ..., t.$

Then experts $E = \{e_1, ..., e_N\}$ are asked to determine their level of agreement on the statements based on a linguistic scale with m terms for every pair (l_i, v_j).

The linguistic terms in the model are represented as fuzzy numbers. In other words we have a mapping $\Phi : T \to F$ from the set of linguistic terms into the family of fuzzy numbers.

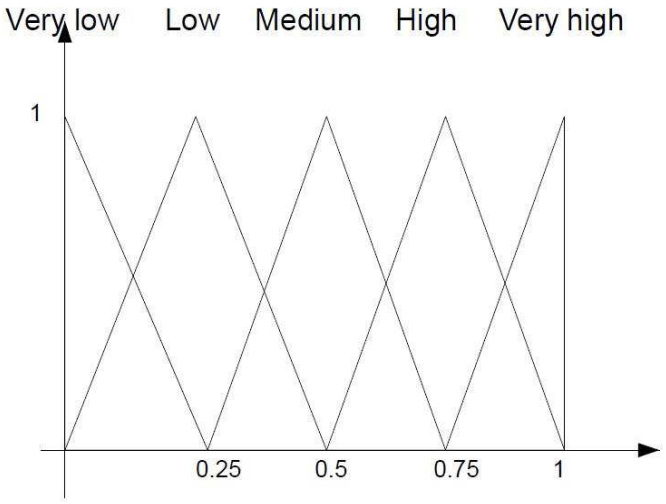

Fig.5. Possible representation with triangular fuzzy numbers

Example 1. One possible representation for a linguistic label is a triangular fuzzy number:

$$A(u)=\begin{cases}1-\dfrac{a-u}{\alpha}, & a-\alpha\le u\le a\\ 1-\dfrac{u-a}{\beta} & a\le u\le a+\beta\\ 0 & \textit{otherwise.}\end{cases}$$

From the opinion of the experts we obtain the frequencies of the different classes.

For the pair (l_i, v_j) we have $n_1^{ij},...,n_m^{ij}$ If we denote by $A_1,...,A_m$ the fuzzy numbers corresponding to the linguistic labels, we can define a new fuzzy number A_{ij} as a "weighted average", with level sets:

$$[A_{ij}]^{\gamma}=\left[\frac{1}{N}\sum_{k=1}^{m}n_k^{ij}a_1^m(\gamma),\frac{1}{N}\sum_{k=1}^{m}n_k^{ij}a_2^m(\gamma),\right]$$

where $[a_1^k, a_2^k]$ is the level set of A_k. This is clearly a fuzzy number with the support in the interval $[0,1]$.

To obtain the connection degree for the pair (l_i, v_j), we calculate the f-weighted possibilistic mean value of A_{ij}, defined in Carlsson and Fuller (2001).

Definition 2. The f-weighted possibilistic mean value of $A \in$ F, with γ -level sets $[A]^{\gamma} = [a_1(\gamma), a_2(\gamma)], \gamma \in [0,1]$, is defined by:

$$E_f(A) = \int_0^1 M(U_\gamma) f(\gamma) d\gamma = \int_0^1 \frac{a_1(\gamma) + a_2(\gamma)}{2} f(\gamma) d\gamma \quad (2)$$

where U_γ is a uniform probability distribution on $[A]^{\gamma}$ for all $\gamma \in [0,1]$.

After we have obtained these defuzzified numbers as the estimation of the connection strengths, we can determine for every attack component the ranking of the other nodes, then we can construct the adjacency matrix of the attack tree by connecting the leaves to the best ranked vertices.

Example 2. In the simplest case we can represent the linguistic labels as a fuzzy set with the membership function

$$A(u) = \begin{cases} 1 & u = c \\ 0 & otherwise \end{cases}.$$

If we have 5 categories, we use the set $\{0,0.25,0.5,0.75,1\}$. The weighs of the outcomes are the frequencies of the linguistic labels. If we observe the weights $n_0, n_{0.25}, n_{0.5}, n_{0.75}, n_1$, then A_{ij} is just the characteristic function of the value

$$\frac{1}{\sum_{j=0}^{4} n_{0.25*i}} \sum_{i=0}^{4} n_{0.25*i} * 0.25i,$$

what is simply the sample mean value of our data. And according to the used defuzzification method, the obtained connection estimation is this sample mean.

4. The architecture of FIDES

In this section we will describe the architecture of FIDES (Fig.6), showing how the main components are interrelated and the role played by the inspectors and auditors, the experts which have to decode the alarms generated by software agents and to detect and describe the suspicious human behaviours (see, e.g. Sanchez et al. 2009 and Edge et al. 2007).

FIDES indeed, has been built on the base of the suggestions we collected interviewing the managers of risk management department of a leading European bank and thus the multi-agent system has been designed according to their opinions.

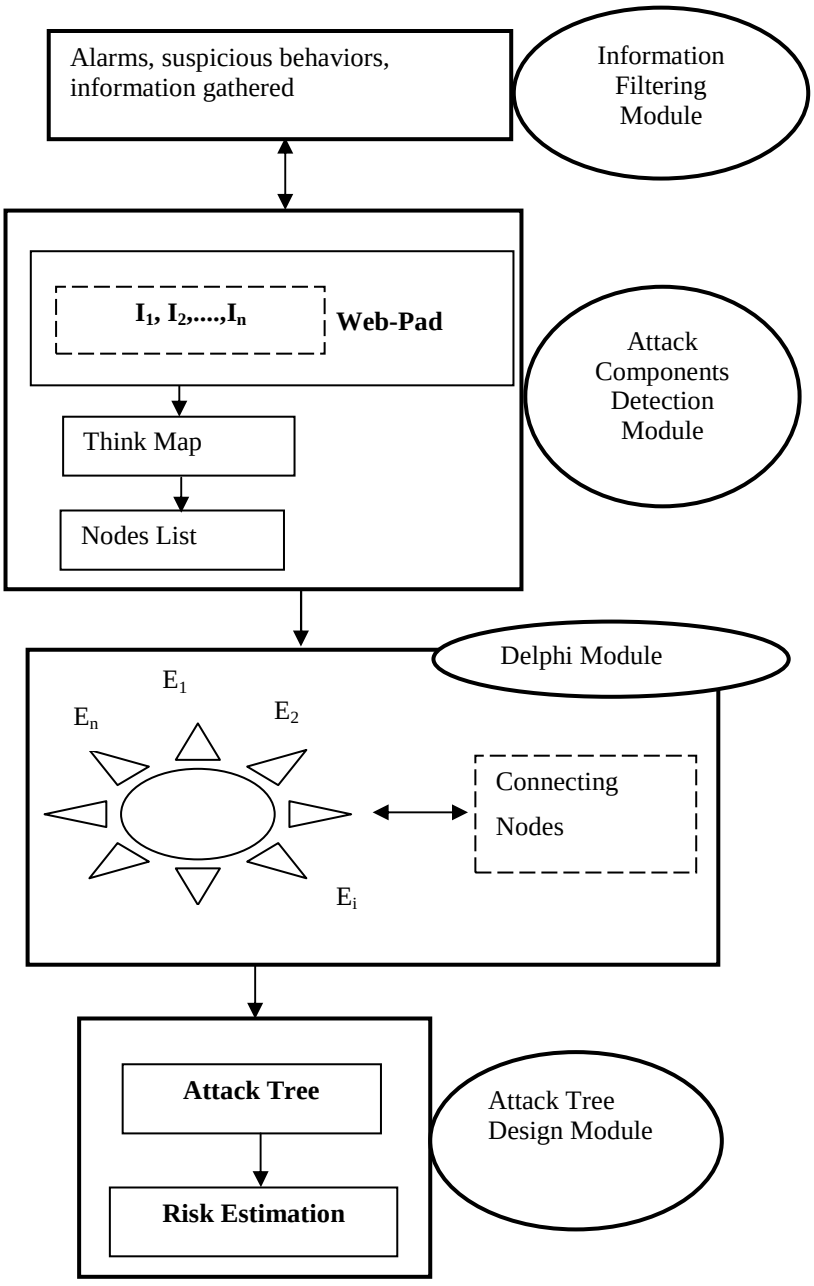

Figure 6. The architecture of the system

In a fraud detection process an audit team has to deal with numerical data and alarms, produced by software agents, but also documents, reports and information gathered by different actors (managers, whistleblowers, anonymous informers) and then summarized by inspectors.

Therefore, the key factor in detecting fraud behaviours is to improve the interaction between inspectors and auditors. Alarms, generated by software agents and/or suspicious behaviours noticed personally by inspectors are filtered using the Web-Pad software. The Information Filtering Module is nothing else that a preliminary session where inspectors can decide which alarms and behaviours to take into considerations to perform all the process. Once the case has been well detailed as shown in Fig.2, inspectors start to underline the keywords in order to create the think-map as described in paragraph 2.

This is the Attack Components Detection Module whose output is the set of nodes to be sent to the auditors.

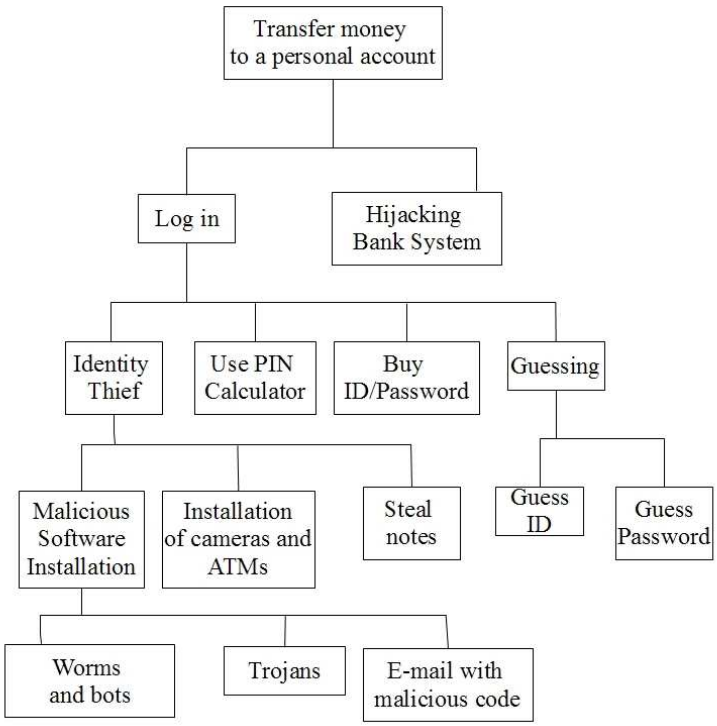

Figure 7. Final attack tree.

Since think-map represents a hypothesis of correspondence between behaviour, suspected fraudsters and fraudulent actions, inspectors create nodes to try to offer an explanation of how this conceptual framework can be associated to concrete actions performed by the fraudster. Nodes are elementary attacks expressed by labels which define all the possible steps of the fraudulent action.

At this stage auditors activate the Delphi Module, driven by the moderator, as explained in paragraph 3, to the end of connecting the nodes to form the attack tree, taking into account the strength of the links between nodes. In the Attack Tree Design Module, experts can estimate the risk and develop the strategy to persecute the fraudster. The estimation process can be performed calculating the most probable or the least expensive path.

Continuing with the example described in paragraph 2.1, in Fig.7 is shown the final attack tree as result of Delphi process where auditors connect the nodes delivered by the inspectors. In this example the goal of the attacker is to transfer money to his/her personal account. The tree shows different ways he/she can achieve this goal. The easiest but the least successful path is to guess ID and password. The most difficult one is to hijack the bank system. The other paths show identity thief sub-attacks based on malicious software installation or the use of cameras installed at ATMs.

In order to select the most probable path, in the risk evaluation phase, experts can have the opportunity to compare the actual attack tree to the ones created in the past and use the information that could be useful to develop a counterstrategy for the new attack. Our assumption is that the identified attack of similar trees can be useful in finding the real path in the present tree. Moreover, by comparing the background of the cases we can obtain useful information for example the possible amount of fraud, the fraudster and/or his/her strategy. A typical situation could be that the same person is performing the same attack, which can be discovered by identifying similar attack trees in a specific range of time.

An attack tree represents the set of potential fraud schemes. After we constructed the tree, we would like to determine the real attack which takes place in the present. To do this, we need a database containing all the attack trees constructed in the past, then comparing the newly created tree with the ones in the database and based on the similar trees and their outcomes, we will be able to choose the most probable attack.

A graph (in our special case a tree) can be represented by its adjacency matrix. The adjacency matrix can be used to find the similarity value between attack trees but the problem is that a tree with n vertices has n-1 edges so the matrix is very sparse. Moreover, since we need comparable matrices for determining the similarity, we have to consider the matrix with all the possible keywords for every tree, and if the number of keywords is k, even in the best case we will have approximately k(k-2) number of 0's in the matrix. Our aim is to find trees with similar sets of edges. It is important to note that every vertex of an the attack tree (like every decision tree) can be associated with the distance value from the root node. Thanks to the structure of the attack tree, we are able to represent the edges of the tree as ordered pairs where the first element is the vertex with smaller distance from the root (the goal of the attack). The set of these pairs is different for two attack trees and if we know this set we can construct the tree. Then we find the similarity of these sets (the number of pairs in a set is equal to the number of vertices in the tree minus 1).

To obtain the similarity of two sets we will employ an index which is frequently used in information retrieval, the cosine similarity. Before the definition of the index, we

need to introduce a notation: if we have to sets, S_1 and S_2 then $M = |S_1 \cap S_2|$ is the number of common items between S_1 and S_2. The cosine similarity of two sets can be defined as

$$\cos(S_1, S_2) = \frac{M}{\sqrt{|S_1||S_2|}}. \qquad (3)$$

where $|S_i|$ is the number of items in S_i. In our case the sets consist of the edges of the trees, so if a tree has n vertices, then the corresponding set will contain n-1 elements. Finding this similarity measure can be done effectively for example by using the method described in Nanopoulos (2002).

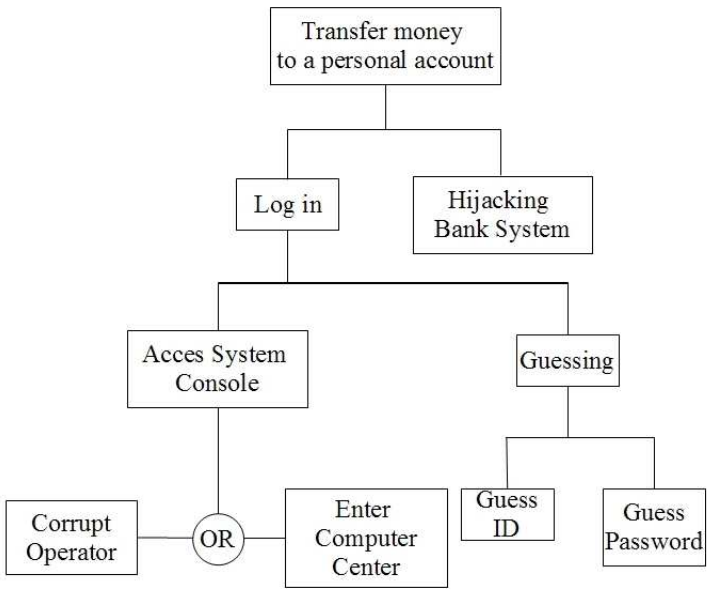

Fig.8. Possible attack tree from the Data Base.

Example 3

If we look at the two constructed attack trees on Fig. 7 (S_1) and Fig. 8 (S_2), we can calculate the similarity by using the formula (3).

The number of edges is 14 and 9 in S_1 and S_2 respectively. If we compare the sets, we can see that the two trees have 5 edges in common: (transfer money to a personal

account, log in), (transfer money to a personal account, hijacking bank system), (log in, guessing), (guessing, guess ID), (guessing, guess password). This means that M=5, and the cosine similarity:

$$\cos(S_1, S_2) = \frac{M}{\sqrt{|S_1||S_2|}} = \frac{5}{\sqrt{9 \times 14}} \approx 0.15.$$

5. Conclusions

The most of the fraud surveys published in the last five years by leading international consulting companies have shown that behavioural aspects play a central role, and therefore the biggest problem to be addressed by auditors and inspectors is to interpret the signals and summarize the information coming from several, sometimes conflicting, sources. The successful solution of this kind of problems depends to a large extent on a proper combination of critical analysis, knowledge-based actions, whistleblowers' messages interpretation, involving groups of interacting experts.

Since before addressing the design of our system we had the chance to meet several experts in charge of diverse risk management activities inside one of the largest European banking group, our work has been inspired by the information collected during the meetings. Banks have a huge amount of data and experience concerning fraud cases, but they don't use it in an efficient way, since the most of knowledge is unstructured. One of the main challenges of the bank is to unify fraud risk assessment processes in the different countries where its branches are located, perform a realistic temporal analysis and establish cause/effect relationship in a rather short time combining objective information with the subjective judgments expressed by experts.

Treasuring the knowledge and the opinions collected during the meetings, we designed the multi-agent system FIDES, intended for the management of fraud detection situations where inspectors and auditors are collaborating according to a two phase detection process. In the first phase, think-maps and Delphi method offer to the inspectors and auditors respectively an effective environment to explicit their knowledge, select the most likely fraud attack components, and finally structuring them in an attack tree. In the second phase, the attack tree structure is definitively settled introducing a representation of uncertainty involved based on fuzzy numbers. Attack tree-based representation of fraudulent processes has the advantage of offering a clear visualization of the attack which permits to the auditors to highlight the most probable attack paths, after introducing the cost or impact of an attack starting from a set of attributes attached to the nodes of the attack tree.

Based on the suggestions and comments collected during the work sessions with the members of the audit team of the bank, we started to develop a first prototype of FIDES. The prototype is limited to two modules, i.e., the Attack Components Detection and Delphi ones. A test has been conducted with the collaboration of the inspectors of the bank to verify the usability and understand the limitations of the system. During the experiments a list of different cases were presented as a short description and possible consequences (typically a loss from a bank account) will be provided to the users (inspectors) in order to create think-maps and then nodes as described in the previous section.

A second group of experts will carry out the process of linking the nodes, aiming at designing the final attack tree. The group process will be driven by a moderator, selected among the members of the same group, according to the Delphi method, and the consensual dynamics based on software implemented in Fedrizzi et al. (2008).

After we have sent few screenshots of the first draft of the prototype, we asked the users to give us a short feedback highlighting positive and negative aspects about what they observed and suggest further developments. The first reaction was a positive evaluation of the improvement of the interaction dynamics between inspectors and auditors. Another feature which has been appreciated is the anonymity, which allowed the users to operate with more freedom, without the pressure of performing a wrong evaluation of the case.

Regarding negative aspects, users pointed out possible problems in the interpretation of the fraud cases once the system was used by people with different cultural backgrounds and languages, a problem arising due to the international profile of the banking group. A limitation would be the possibility to manage big fraud operation on international scale, where different auditors and inspectors from different contexts might find difficulties in understanding each other. In particular in building the think-map, the perception and interpretation of the facts might encounter problems on a semantic level, like underestimate or overestimate the same behaviour or action performed in different contexts.

Concerning future improvements of FIDES, first of all, assuming that the formally expressed representation of the information related to the attacks (attack trees) is stored in references sources (catalogue of attack trees), an intelligent searching module will be introduced for retrieving information from the catalogue. To wit, our aim is to propose an ontology-based description of the attack trees to provide a formal basis for sharing knowledge and for reusing it during the attack tree design process.

Another possible improvement would be a mobile version of FIDES, particularly useful when inspections are carried out in the different branches of the bank spread to a large geographic area, and aiming at homogenize the information coming from different information sources.

References

1 Bazerman, M., Lowenstein, G., Moore, D., 2007, Why good accountants do bad audits. *Harvard Business Review*, 3-8.

2 Basel II, 2006, *Basel Committee on Banking Supervision*.

3 Buoni, A., 2010, Fraud detection: From basic techniques to a multi-agent approach. *International Conference on Management and Service Science*, Wuhan, August 24-26.

4 Buzan, T., 1974, *Use your head.* (London: BBC Books).

5 Carlsson, C., Fedrizzi, M., Fuller, R., 2004, *Fuzzy logic in management*, (Dordrecht; Boston ; New York, Kluwer Academic).

6 Carlsson, C., Fuller, R., 2001, On possibilistic mean value and variance of fuzzy numbers. *Fuzzy Sets and Systems*, 122, 315-326.

7 Chou, C.L-Y, Du, T., Lai, S. V, 2007, Continuous auditing with a multi-agent system. *Decision Support Systems*, 42 (4) 2274-2292.

8 Dubois, D., Prade, H., 1978, Operations on fuzzy numbers, *International Journal of Systems Science*, 9 (6), 613-626.

9 Edge, M.E., Sampaio, R.P.F., Choudhary, M., 2007, Towards a proactive fraud management framework for financial data streams. *Proceedings of the Third IEEE International Symposium on Dependable, Autonomic and Secure Computing*, 55-64.

10 Fedrizzi, M., Fedrizzi, M., Marques Pereira, R., Brunelli, M., 2008, Consensual dynamics in group decision making with triangular fuzzy numbers. *Proceedings of the 41st Annual Hawaii International Conference on System Sciences*, IEEE Computer Society, 70 -78.

11 Fedrizzi, M., Kacprzyk, J., Nurmi, H., 1997, *Consensus under fuzziness*, (Dordrecht; Boston ; New York, Kluwer Academic).

12 Gordon, T. J., 1994, The Delphi method in futures research methodology. *AC/UNU Millenium Project*, Washington, AC/UNU.

13 Grazioli, S., Johnson, P. E., Karim, J., 2006, A cognitive approach to fraud detection. http://ssrn.com/ abstract= 920222.

14 Hand , D. J., 2007, Statistical techniques for fraud detection and evaluation.

http://langtech.jrc.ec.europa.eu/mmdss2007/htdocs/Presentations/Docs/MMDSS_Hand_PUBLIC.pdf.

15 KPMG Fraud survey, 2009.

16 KPMG Forensic fraud barometer, 2009.

17 Ma, Z., Shao C., Mac, S., Ye, Z., 2011, Constructing road safety performance indicators using Fuzzy Delphi Method and Grey Delphi Method, *Expert Systems with Applications*, 38, (3), 1509-1514.

18 Mauw, S., Oostdijk, M., Foundation of attack trees, 2005, Information security and cryptology. *Proceedings of the 8th Annual International Conference on Information Security and Cryptology*, 186-198.

19 McElhinney, E., 2010, Factors which influence nurse practitioners ability to carry out physical examination skills in the clinical area after a degree level module – an electronic Delphi study, *Journal of Clinical Nursing*, 19 (21-22), 3177-3187.

20 Nanopoulos, A., Manolopoulos, Y., 2002, Efficient similarity search for market basket data. *The VLDB Journal*, 11, 138-152.

21 Niitsoo, M., 2010, Optimal Adversary Behavior for the Serial Model of Financial Attack Trees, *Proceedings of the 5th International Workshop on Security*, (LNCS 6434, Springer-Verlag Berlin Heidelberg), 354–370.

22 Novak, J., Cañas, A., 2006, The theory underlying concept maps and how to construct them. *Technical Report IHMC Cmaptools*.

23 Oxman, R., 2004, Think-maps: teaching design thinking in design education. *Design Studies*, 25 (1),63-91.

24 Rowe, G., Wright, G., 1999, The Delphi technique as a forecasting tool: issues and analysis. *International Journal of Forecasting*, 15, 353-375.

25 Sanchez, D., Vila, M.A, Cerda, L., Serrano, J.M., 2009, Association rules applied to credit card fraud detection. *Expert Systems with Applications*, 36, 3630-3640.

26 Schneier, B., Attack trees. *Dr. Dobb's Journal* December 1999, available at http://www.schneier.com/paper-attacktrees-ddjft.html.

27 Steinert, M., 2009, A dissensus based online Delphi approach: An explorative research tool, *Technological Forecasting & Social Change* 76, 291-300.

28 Wang, D.G., Li, T., Liu, S.J.L, Liang, G., Zhao, K., 2008, An immune multi-agent system for network intrusion. *Proceedings of the third International Symposium on Intelligence Computation and Applications*, (LNCS 5370, Springer-Verlag Berlin Heidelberg), 436-445.

29 Zhang, L.S., Zhou, N., Wu, J.X., 2008, The fuzzy integrated evaluation of embedded system security. *International Conference on Embedded Software and Systems*, 157-162.

30 Åhlberg, M., 2007, History of graphic tools presenting concepts and propositions. http://www.reflectingeducation.net/index.php?journal=reflecting&page=article&op=downloadSuppFile&path%5B%5D=49&path%5B%5D=6.