DYNAMIC ONTOLOGIES IN INFORMATION SECURITY SYSTEMS[1]

Vladimir Jotsov

Abstract: *Different types of ontologies and knowledge or metaknowledge connected to them are considered and analyzed aiming at realization in contemporary information security systems (ISS) and especially the case of intrusion detection systems (IDS) or intrusion prevention systems (IPS). Human-centered methods INCONSISTENCY, FUNNEL, CALEIDOSCOPE and CROSSWORD are algorithmic or data-driven methods based on ontologies. All of them interact on a competitive principle 'survival of the fittest'. They are controlled by a Synthetic MetaMethod SMM. It is shown that the data analysis frequently needs an act of creation especially if it is applied to knowledge-poor environments. It is shown that human-centered methods are very suitable for resolutions in case, and often they are based on the usage of dynamic ontologies .*

Introduction

Contemporary ISS and especially the web-based systems are primarily using intelligent methods. IDS or IPS are machine learning oriented, and some of them are using knowledge discovery and data mining [1,2]. Such sophisticated technologies are time- and labor-consuming, and it is very hard to make them satisfy the standard demands for convergence of the results and/or comparatively low computational complexity. However designers and customers accept such difficulties trying to gain from higher reliability of such applications. The base concept of the presented paper is to make a powerful human-centered system combined with firewalls, IPS or other security tools. It could make a complex defense against different groups of intruders. Aiming at that, we should introduce different ontologies to support the IDS work or the system will be not enough reliable. In the next two sections we'll show the usage of ontologies in different decision support methods and applications in data mining, web mining and/or other computation discovery or evolutionary systems.

Usually ontologies are issued to support methods or applications to probabilistic, fuzzy inference or uncertainty processing [3-6]. Our research shows [7] other, nonstandard ways that are not excluding the other contemporary research but are making something in addition to well known methods, and so are useful to be combined with. The next Section is dedicated to a new self-learning method that constantly searches for knowledge conflicts or its ultimate case-contradictions-and tries to resolve them [8]. It is found to be the best way to self-improvement via the constant correction of knowledge incompleteness or inconsistency. On contrary to other machine learning methods, our proposal is ontology-driven and it is much less heuristic by nature than the other well known methods from the field. In this case the keyword self-learning is introduced to emphasize the above quoted differences.

In section 3 different human-centered methods are used to check the truth value of one or group of statements. Those statements are named below in the text: definition of the problem, target question or a *goal* for short, e.g. goal to detect a possible intrusion. We are not trying to elaborate completely automatic systems. Since the first knowledge discovery systems, it is seen that making more or less automatic inference system makes it over fulfilled of heuristics which restricts its future development. Instead we offer making the machine the best human's advisor. The machine founds some interesting patterns and represents it in an user-friendly manner. Some of similar ideas are used in cognitive graphics but our methods are absolutely different and we prefer to name the field: *human-machine creation*. It is shown that in many cases the act of creation isn't something very difficult, it may resemble a human-machine brainstorm method where the machine 'mechanical' part of work aims at catching some repeating/resembling patterns or show other relations or regularities to the user who may take his fuzzy part of the investigation.

[1] The paper is supported by NSF grant 'Dynamic Ontologies in Multiagent Information and Control Systems'– Bulgarian Ministry of Education and Science

Application results are considered in section 4. Never the above quoted research has been realized in 'all in one' system because of its high complexity. However we used a big variety of method combinations under the SMM, synthetic metamethod control. Those allow us make rather effective inference machines.

2. Ontology-Based Machine Learning

Let the strong (classical) negation is denoted by '¬' and the weak (conditional, paraconsistent [9]) negation by '~'. In case of an evident conflict (inconsistency) between the knowledge and its ultimate form–the contradiction–the conflict situation is determined by the direct comparison of the two statements (the *conflicting sides*) that differ one form another just by a definite number of symbols '¬' or '~'. For example, A and ¬A; B and not B (¬ is equivalent to 'not'), etc. η is a negation type, in case strong classical negation, and square brackets embrace all possible words used to represent explicit strong negations in texts.

$$\{\eta\}\ [\text{no, not, не, нет}]. \qquad (1)$$

The case of implicit (or hidden) negation between two statements A and B can be recognized only by an analysis of a present ontologies of type (2).

$$\{U\}\ [\eta\text{: A, B}]. \qquad (2)$$

where U is a statement with a validity including the validities of the concepts A and B and it is possible that more than two conflicting sides may be present. Below it is accepted that the contents in the figure brackets U is called *an unifying feature*. In this way it is possible to formalize not only the features that separate the conflicting sides but also the unifying (or common) concepts. For example the intelligent detection may be either automated or of a man-machine type but the conflict cannot be recognized without the investigation of the following conflict ontology (3).

$$\{\text{detection procedures}\}\ [\neg\text{: automatic, interactive}]. \qquad (3)$$

Ontologies (1) or (2) describe situations where conflict the sides mutually negate one another. In the majority of situations the sides participate in the conflict only under definite conditions: $\chi_1, \chi_2, \ldots\chi_z$.

$$\{U\}\ [\eta\text{: } A_1,A_2,\ldots A_p]\ \ <\tilde{\chi}_1 * \tilde{\chi}_2 * \ldots * \tilde{\chi}_z>. \qquad (4)$$

where $\tilde{\chi}$ is a literal of χ, i.e. $\tilde{\chi}\equiv\chi$ or $\tilde{\chi}\equiv\eta\chi$, * is the logical operation of conjunction, disjunction or implication.

Let the ultimate form of conflict, contradiction is investigated. The syntactic contradiction ontology is depicted in fig. 1, and the semantic variant is considered in fig. 2. It is obvious that the contradictions are very different but their base ontologies seem quite similar. The reason is that the essential part of both conflicts or contradictions from (2) and (3) isn't an ordinary ontology knowledge itself but is a form of *metaknowledge* that controls the usage of ontologies or parts of them. The bottom level objects from fig. 1 unconditionally refute each other. We may find some cases where the same system have been automatic one, and after some time it became an interactive system, but this case is so labor consuming that actually we speak about a new, different system.

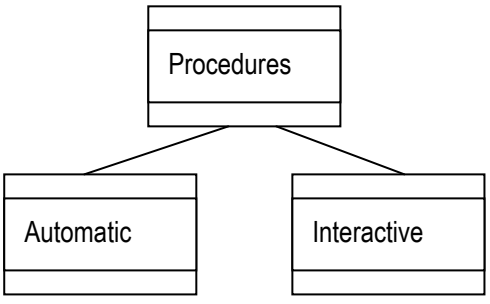

Figure 1. Ontology for a syntactic contradiction

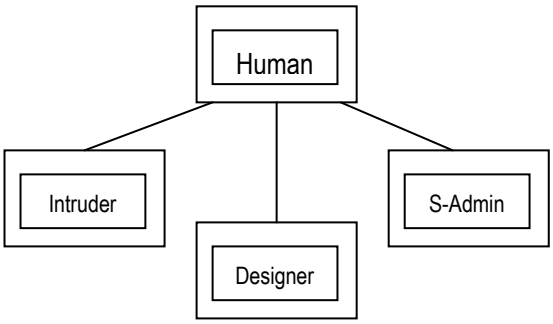

Figure 2. Ontology for a semantic contradiction

What is depicted in fig. 2 shows a different situation, concerned with 'IDS-humans' or three major groups of people dealing with IDS: intruders; security experts or designers (designer); security administrators (S-admin). Weak negation is used in case, in the bottom level objects, because the security administrator may be former

expert or he may be a designer of another system, and also former hackers may be engaged as experts. The semantic contradiction will appear iff all the following conditions are satisfied: T (the same time) and I (the same system) and U (the same person) and P (the same place). Next figures 3 and 4 give more details for the case.

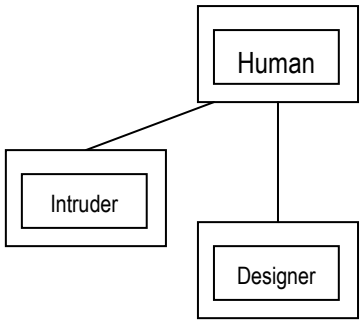

Figure 3. Ontology for conflict situations

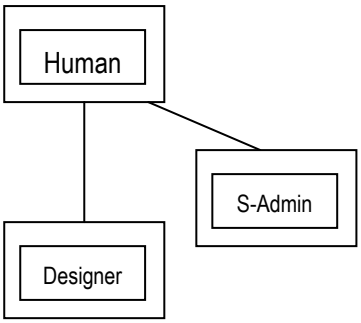

Figure 4. Ontology for contradiction situations

Fig. 3 concerns the part of ontology from fig. 2 when the security administrator is eliminated. Let all the quoted above conditions are satisfied: T (the same time) and I (the same system) and U (the same person) and P (the same place). Still we couldn't define the situation in fig. 3 as a contradiction, say because the designer may test the IDS system. To resolve this situation we may use knowledge type exclusion and defeasible inference or other well known inference schemes. This is an example of knowledge conflict, not a contradiction, and only additional investigations may result in semantic contradiction.

Fig. 4 shows the semantic contradiction, and if the conditions $T \wedge I \wedge U \wedge P$ are satisfied, then the contradiction appears: 'nobody can occupy both positions'. Thus fig. 2 contains different types of ontology knowledge inside it. The above given examples aim to show that processes based on knowledge conflicts or contradictions couldn't be thoroughly described by an ontology knowledge, and using only static situations. We need the dynamic picture to decide if we have no conflict or contradiction. On the other side, when the situation dynamics is investigated, pretty often we turn on to ontology corrections due to its incompleteness or incorrectness. In this situation the main conclusion for us is the following. We need to use metaknowledge and dynamic ontologies to cope with conflict or contradiction identification. The conflict identification is almost always much more complicated than the contradiction case.

The ontology-based contradiction identification is followed by its resolution [8]. The proposed resolution methods are applications of ideas from nonclassic logics and they are one of base parts of the presented research in analogy inference machines, case-based methods, data mining, etc. Contradiction resolution depends on the situation and types of contradiction sides. Our research [8] revealed five main groups of resolution scenarios. Currently we make investigations to elaborate new contradiction resolution scenarios. The research shows that automatic contradiction resolution processes may stay active constantly using free computer resources. Also they may be directly activated by user. In the first case the knowledge and data bases will be constantly improved by continuous elimination of incorrect information or by improving the existing knowledge as a result of revealing and resolving contradictions.

Only two-sided contradictions are considered because most of multi-sided contradictions are represented as a set of two-sided contradictions. The automatic contradiction resolution process starts when one of the sides is very weak, say machine hypothesis while the other side is rather strong, say expert knowledge. The resolution finishes with an instant elimination of the weak side. This situation is used to filter machine hypotheses. If we use the notion 'conflict' instead of 'contradiction', then in the same situation analogically another filter is to be built up.

Another automatic resolution process is where no resolution is needed at all. Say, if the task is to know the speed of light, then we don't need to resolve the contradiction 'is the light wave or particle?' This way is widespread in multi-agent systems. It is supposed the agents may return without collisions to previous positions before the conflict. Whenever possible, we use each possibility to escape from contradiction resolution process but each detection of conflicts/contradictions should be alerted.

A third group of methods for automatic resolution is the following. If {P} is a set of parameters in the considered model, and every $p \in P$ is strictly defined: $a \le p_i \le b$, and the considered value is outside the model range, say $p_i=b+10$, then there exists a contradiction with the model, and this contradiction is simply resolved by issuing a warning: 'p_i exceeds the range limit'. After that the model and/or some factors connected to the considered parameter should be corrected. Of course the above considered example could include ontologies and nonnumeric information, say 'we thought you are using statistical methods and they aren't'.

All the other ways use human-centered methods where an expert or even non-advanced user contribute to the resolution process. The machine prepares all the necessary information including all the ontologies involved with some additional features, say dynamics of their changes. If an inference to the sides of the contradiction exists then the inference tree is represented. All knowledge and data is to be represented using below considered CALEIDOSCOPE method. The purpose of this group of methods is to reveal hidden regularities to the user and to group all the information so that to ease his act of creation.

As a result the considered contradiction resolution methods have been upgraded to a machine learning method i.e. learning without teacher which is rather effective in case of ISS.

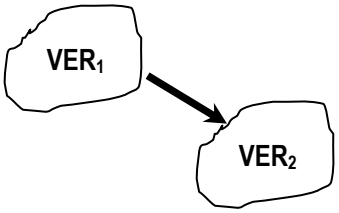

Figure 5. Standard way to changing ontologies.

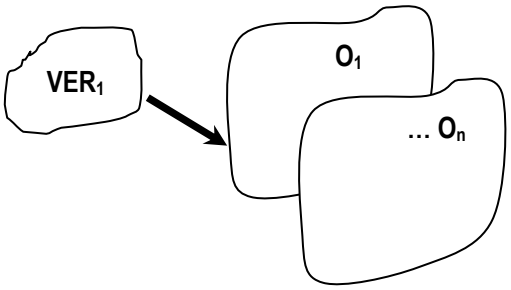

Figure 6. Changing and defeating ontologies.

Ontologies involved in the contradiction resolution process are divided in the following two groups. Denote $o^*_i \in O^c$ are ontologies from the model of syntactic/semantic contradiction, $obj_S(o^*_i)$ are objects from all the ontology levels from O^c and $o_i \in O$ are all ontologies concerning the sides of the contradiction. Let neg(a,b) denote the arguments of neg unconditionally refute each other or they refute each other while all the conditions are executed. If $neg(o^*_U , o^*_V)$, then the set O^c should be altered by addition and/or elimination of its elements. On the other hand, O^c contains ontology descriptions linked to the ontology from square brackets in the left hand side of (1) up to (4). Hence the contradiction resolution process leads to dynamic ontologies in the knowledge base (**KB**). If a negation is revealed between objects from different levels of one or different ontologies: $neg(obj_S(o^*_I) , obj_S(o^*_J))$, then it doesn't mean the contradiction exists but the situation should be saved and described to the domain experts. Even if no contradiction, it may shift the ontologies involved. If $neg(o_S , o_T)$, then the contradiction exists but the models of contradictions O^c are not the reason for its appearance. In this case other domain knowledge including ontologies $o_i \in O$ should be altered and/or appended which may lead to changes in existing ontologies. The existing software, say RDF, OWL or Protégé, contains standard set of features to shift/append ontologies while keeping its history [10,11]. The process of changing ontologies is depicted in fig. 5 where ontology version 1 (ver_1) is substituted by ver_2 but the transition from version 1 to version 2 could be checked at any time. On contrary to the traditional scheme, we offered in [12] a defeasible inference scheme with the following outcome. The original ontology (ver_1) will be substituted by one or a set of ontologies using exclusions while its (ver_1) is still valid in an a priory given possible world. Thus the revealed contradiction is resolved by the transfer of the inconsistent sides or derivatives to different possible worlds.

Example.

Let an ISS is applied into a sport environment and following groups of people may be involved in the training process of the security administrator.

{training of security administrators} [~: trainer, instructor, manager/policymaker, developer] $< T \wedge I \wedge U \wedge P >$. (5)

If the environment is shifted from sports to government, then instead of trainer an expert will be used and the model will be as follows.

{training of security administrators} [~: expert, instructor, manager/policymaker, developer] < T∧I∧U∧P>. (6)

Thus an invariant (*strong*) part of the set is observed which isn't changed by shifting environments, and the difference is the following. In sport environment the meaning sports expert is defeated by trainer. Models (5) and (6) are correctly functioning in *different possible worlds* and there is no inconsistency because of shown differences between (5) and (6). Metaknowledge is used to address (5) to sports and (6) to government.

Designers are seldom involved in the administration training. When the sport environment is shifted by another conditions, say (6), it may be seen that the above written is the same everywhere. Thus the designer branch from the set of the involved ontologies from (5) or (6) is to be pruned, making the work with ontologies more effective. Hence the higher quality of ontologies is obtained. The same manner we may decrease some ontology levels using dynamic transfers like (5) to (6).

Generally, the ontologies dynamics from O^c is is based on contradictions resolution which leads to changes in the set or defeasible inference attachments to the set [12]. After that the quality of ontologies from O^c grows higher because they are becoming less incomplete or incorrect, and because ineffective branches from the set have been pruned if necessary.

To conclude this section, best ontologies used in contradiction resolution processes are dynamic ontologies. The resolution is an evolutionary process [13] and it brings dynamics to knowledge involved, leading to new forms for ontology processing. Two contemporary concepts may be shown how to make machine self-improvement leading to self-learning. The first one is based on the usage of artificial neural networks (ANN), or similar heuristic methods. The ANN methods show low learning rate and high design costs. On contrary, we offer machine self-improvement via contradiction or knowledge conflict resolution. KB is improving after every resolution process, and this gives dynamics to ontology descriptions. After the resolution, the *invariant* part of knowledge or method remains that makes it stronger and more flexible. This self-improvement needs only one time-consuming resource: juxtapositions between different groups of knowledge. It needs the human help only in some complex situations. The considered machine learning is an evolutionary process [13] and it gives better results if the intermediate solutions (*hypotheses*) are tested in different models [14]. The system has many resources to constantly resolve the contradictions when no goal is given or in parallel to main jobs. We can't escape from heuristics but they are passed to the decision maker via productive human-machine interactions mechanism thus making the system alone more effective and less complex. Some part of heuristics is hidden in ontologies driving the process of learning. Most of the presented computational discovery/data mining methods are data-driven. The considered research is more ontology-driven than data-driven but it belongs to the same group of methods. The below presented methods allow us to use not only statistical methods but also other knowledge acquisition methods for knowledge discovery.

This type of machine learning is novel and original in both theory and applied aspects.

3. Method Interactions under SMM Synthetic Metamethod Control

The described below methods interact under the common control of a new type of a synthetic metamethod (*SMM*). The considered metamethod avoids or *defeats* crossovers, phenotypes, mutations, or other elements from traditional evolutionary computation [13, 15]. The formal description below is appended with few explanations in an analogous manner as the way to reduce extra descriptions, because the general scheme of the chosen strategy is rather voluminous. *SSM* swallows and controls the following methods:

I. **INCONSISTENCY**: contradictions detection and resolution method;

II. **CROSSWORD** method;

III. **FUNNEL** method;

IV. **CALEIDOSCOPE** method.

A. CROSSWORD Method

Let somebody tries to solve a problem with a complex sentence of 200+ letters with vague for the reader explanations. Let the unknown sentence be horizontally located. The reader can't solve the problem in an arbitrary manner, because the number of combinations is increased exponentially. Now it is convenient to **facilitate** the solution by linking the well known to the reader information with the complex one from the same model. The reader tries to find vertical words that he is conscious about, say 'non-stream ciphers' (=block ciphers). The more crossings lead to the easier solution of the horizontal sentence. The approach for the CROSSWORD is *even easier*. Here both the easy meanings and difficult ones are from one domain, therefore an additional help exists to find the final solution.

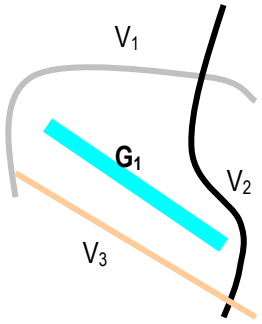

Figure 7. Example of nonlinear constraints

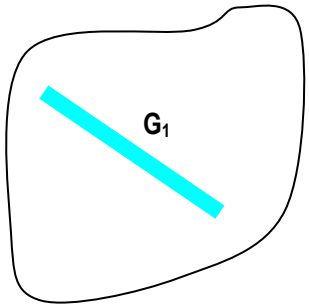

Figure 8. The goal is inside an ontology

The difference of the CROSSWORD method from the usual crosswords is in its highly dimensional spaces and of course the analogy is rather far and is used only for the sake of brevity. Let G be a goal that must be solved, and it is decomposed into two types of subgoals: G_2 is deduced in the classical deductive manner using Modus Ponens (MP), and G_1 is explored in the area defined by the constraints V_1-V_2-V_3 in fig. 7.

The constraints V_j are not necessarily linear. Nonlinear V_j are depicted in fig. 7. Let all the constraints are of different types. Denote V_1 is a curve dividing two groups: knowledge inconsistent with G_1 is located above V_1 and consistent knowledge is below the curve. Let V_1 divides the knowledge having accordance to G_1 from knowledge conflicting with the subgoal. In the end, let V_3 is a linear constraint e.g. x>1997. The solution to the subgoal lays inside the area depicted in fig. 7 and the goal resolution complexity falls significantly.

Another situation reducing the resolution process is depicted in fig. 8 where the same subgoal G_1 is located inside and ontology which gives the search constraints. Sometimes the proof leading to the situation in fig. 8 is the proof *on contrary* when it is impossible the goal to be outside the considered ontology. Comparisons between two examples from fig. 7 and fig. 8 show that using ontologies to reduce the research area is more natural way and is much more effective than standard constraint satisfaction methodology.

Let subgoal G_1 is indeterminate or it is defined in a fuzzy way. Then the introduced algorithm is defined in the following way.

$$\begin{gathered} K_i \in K,\ i=1,2,\ldots n:\ G_1 \cap K_i \neq \varnothing; \\ L_j \in L,\ j=1,2,\ldots m:\ G_1 \cap L_j = \varnothing; \\ S=(G_1 \cap K_1), T=(G_1 \cap K_n); S \neq T; x_1,y_1,z_1 \in S;\ x_2,y_2,z_2 \in T; \\ \frac{x-x_1}{x_2-x_1}=\frac{y-y_1}{y_2-y_1}=\frac{z-z_1}{z_2-z_1} \end{gathered} \tag{7}$$

where x_1, y_1, z_1 and x_2, y_2, z_2 are the coordinates of the respective boundary points from S and T from the set K whilst x, y and z are the coordinates of the points from the slice that tethers the explored area. In this way (by two sticking points) the goal search is restricted from an infinite space to a slice in the space. The introduced method is realized in an iterative manner: the goal place from (7) is replaced by K_i from the previous iteration and so on.

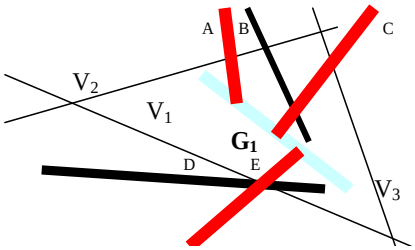

Figure 9. CROSSWORD method: constraints and binding to G

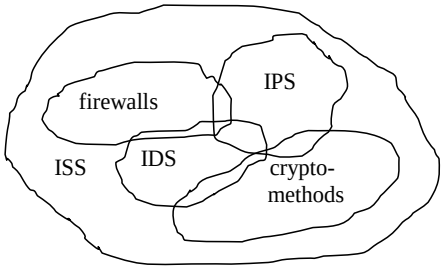

Figure 10. 'Fuzzy' intersections

Fig. 9 illustrates an example with three elements of K={A,B,C} where L={D,E} contains two elements. The example illustrates the benefit from the elements of L and from the spatial constraints V_u even in the case n>1. It is conspicuous that the direction of G_1 most often does not predetermine the integral decision and that the elements D,E and V_u decrease the number of the concurrent alternatives.

Different types of connections are depicted in fig. 9. The search restriction is done by V_1, V_2, and V_3 as considered above in fig. 7. The constraints B and D also restrict the search for G_1 but this restriction is dot-shape because B and D lay not in the search area bounded by V_j. On the other hand, those dots make tight fixation to G_1 , so the are denoted fixation constraints. In the end, A, C and E are resolution constraints because they intersect G_1 and give us parts of the solution to the problem.

The CROSSWORD purpose is to bind and unknown with the known knowledge. Ontologies have been used aiming at realization of different constraints (fig. 8 and fig. 9) and binding elements (fig. 9). Each ontology substitutes a complex set of (non)linear constraints, say in an ontology intrusion which varies in different situations. Thus complex logic and qualitative calculations have been substituted by a keyword/meta tag search. What is depicted in fig. 10 is an application result of methodology for defining inexact borders between few ontology examples. Indeed, if only a little part of IDS is given, then it may be mistaken with a part from IPS or other ISS. The fuzzy ontology border is limited by a knowledge area in the form 'this is exactly not that ontology'. If the system is passive then it is a firewall or similar but not IDS.

One of the intersections from fig. 10 is shown in details in fig. 11. Dynamic ontologies are well suited for operating with such imprecise information. If a contradiction appears then it should be resolved as shown in previous section.

The intersections in fig. 9 are parts of the considered goal from the ontology. The binding element, say B in fig. 9, is knowledge concerning the goal, in other words this knowledge enlarges the belief that the goal is true. In dynamic ontologies this type of knowledge is a target to be included in further ontology versions.

Example.

A security administrator of IPS or IDS receives an email offering perspective positions in the field. He replies and receives large files explaining job offers, and is invited to describe all his project activities. Soon after that the system alerts an increased number of false alarms.

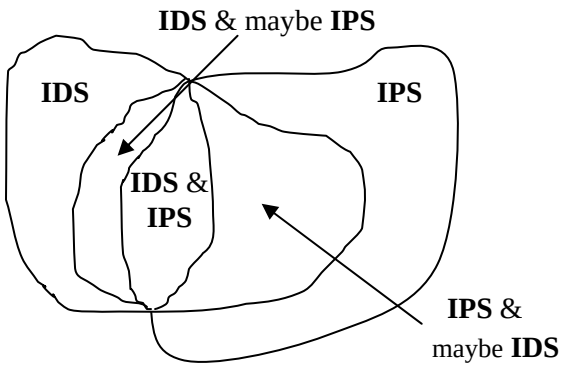

Figure 11. Different grades of intersection.

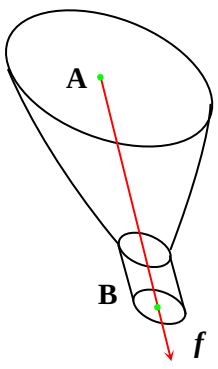

Figure 12. FUNNEL method.

The ontology used here is a hoax. Coincidences in receiving job offer and large files are analogical to binding elements B or D in fig. 9. They don't contain pieces of the goal but are its close neighbors. Large files in this case may purpose preparations to larger traffic, drone software, influencing the artificial neural network from ISS or say ISS shutdown because of false alarms appeared it time of critical work. Intruders are trying to force the administrator to lower the system sensitivity or to turn off the alarms. It is only one intersection with the goal in case: an increasing number of false alarms but because of the other information it is enough to issue an alert and investigate for possible intrusion.

Example

After a visit to some of non-trusted sites, security administrator activates few spyware programs but all of them lead to system shutdown. The administrator switches off the network cable, and observes during the next reset a trial to connect to an unknown .org site. Next few days this PC is working offline and waiting for new Windows installation. After that and before any changes, the spyware shows no threads, the computer functions normally and only the browser is turning on very slowly.

If no dynamic ontologies are used then no threat will be detected. The well investigated situation from the past is connected to present fuzzy situation in case Past knowledge will force further investigations, and new system installs.

B. FUNNEL Method

We denote with $f(t_0)$ a fitness function in the point t_0. In the common case $f(t_0)$ may vary according to its environment – the position in the space and other impacts over the point. In this paper the function is linear and it does not change in the whole domain, $f=f(t_0)$. In this way $f(t_0)$ is reduced to a free vector f. Let $f(t_0)$ is one of the intermediate solutions to the goal when the process has reached up to t_0 . $f(t_0)$ points only to the *recommendable* direction for the evolution of the solution [13], so the movement in this direction shall be realized only if there are no other alternatives. Here we may use a 'gravity' analogy: it is too weak in case of e.g. jets but still it is enough strong not to be underestimated. $f(t_0)$ is combined with a system of spatial constraints in the following way:

$f(t_0)$ is the goal function;

$f_i(t_0)$ is a set of functions which affect t_0.

$$A\frac{d^n x}{d^n t}+B\frac{d^n y}{d^n t}+C\frac{d^n z}{d^n t}\le D \tag{8}$$

$$Ex+Fy+Gz\le H \tag{9}$$

where (8) is a system of non-linear constraints and (9) is a system of linear constraints. Then the direction of the solution in $f^*(t_0)$ is defined as a sum of the vectors multiplied by the respective coefficients k_i; the existing system of constraints is presented by (8) and (9).

$$f^*(t_0)=f(t_0)+\sum_i k_i f_i(t_0) \tag{10}$$

Let's assume you have a *plastic funnel*. If you fix it vertically above the ground, you can direct a stream of water or of vaporous drops etc. If you change the funnel direction, then the stream targeting will be hampered, if the stream hasn't enough *inertia power*. Fixing the funnel horizontally makes it practically useless. Analogically in the evolutionary method the general direction in numerical models is determined likewise. In other words this is a movement along the predefined gradient of the information. Just like in the case of the physical example, there are lots of undirected hazardous steps towards conclusions and hypotheses in the beginning.

This paper offers the following modification of FUNNEL. Let k_i be not constants:

$$k_i(t_0)=\frac{k_i^0}{1+D_0-D} \tag{11}$$

where k_i^0 are the initial meanings coinciding with k_i from (10) and (t_0) are the respective coefficients in point t_0, D is the initial point in the investigated domain–a beginning of the solution and D_0 is an orthogonal projection of t_0 upon the straight line L parallel to f where $D\in L$. In this case moving away from the beginning D the solution depends more and more on the fitness function but the other external factors influence it less and less.

The FUNNEL method can be indirectly based on inconsistency tests with known information. The method may be used also in the other parts of *SMM*, e.g. in the CROSSWORD method it assists the determination of the direction of the explored goal. The graphical representation of the FUNNEL main idea is represented in fig. 12.

It is a data driven method, so intruders haven't possibility to predict the results. The direction *f* from the figure is the goal, e.g. the fitness function from genetic algorithms. Unlike the other contemporary methods, the FUNNEL method gives the ISS freedom to choose and update the hierarchy of goals. In 'the loose part' A in fig. 12, if a new goal appears and promises large gains, and if there is still a long way to resolve *f*, then ISS will try to reach the nearest goal, after that it will return to its way for *f*. The 'edge' constraints in FUNNEL are function of the following parameters: the 'stream inertia' of the intermediate solutions, 'gravity', etc. The next Sections show that INCONSISTENCY method also can be applied to define constraints in the FUNNEL method.

C. CALEIDOSCOPE Method

The CALEIDOSCOPE is the visualization method: it presents the current results or the solution to the security expert. Apart from other interfaces, here some cognitive elements have been applied that help the user make conclusions using notions still unknown to the machine: 'beauty', 'useful', etc. Here the system role is mainly to inspire the decision making imagination and to give him the interesting results: repetitive patterns, etc.

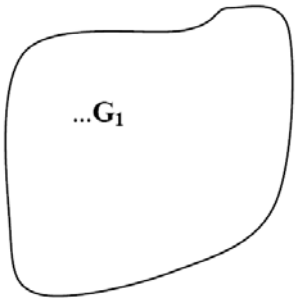

Figure 13.
CROSSWORD -1.
Location of the goal

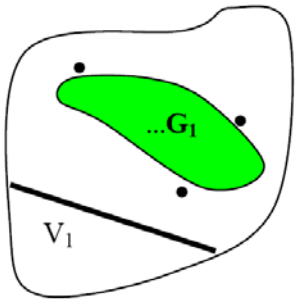

Figure 14.
CROSSWORD -2.
Two different constraint types

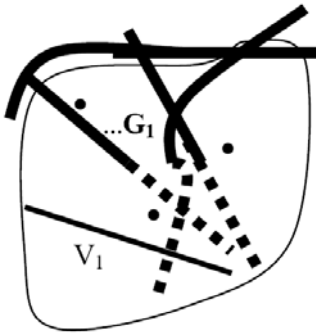

Figure 15.
CROSSWORD -3

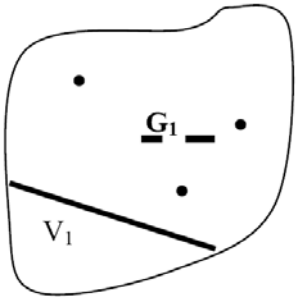

Figure 16.
CROSSWORD -4

Many of the above described methods contain enough visualization elements; in these cases the CALEIDOSCOPE method makes only graphic interpretations of results. In other cases it should make an optimal rotation of the pattern or show intersections of pattern or make other processing helping the user make the decision in best comfort conditions.

Fig 13 shows an example when the decision to the goal is located in the depicted ontology area, and all the other domain knowledge may be considered only if has some relation to the ontology. Let restriction constraint V_1 from fig. 14 is found e.g. 'show only new results', and three fixation constraints are found: the intersection of the curves with the ontology field is represented as three dots. Both two types of constraints make rough solutions thus helping to restrict the search area and make the method complexity better.

Fig. 15 depicts same constraints and three resolution constraints making intersections with the desired goal G_1. A part of other constraints helping define the three resolution constraints is depicted. It is shown in Fig. 16 that two right intersection parts are joined, and the left part is enlarged using knowledge modeling, binding and logic methods. Thus a big part of the goal is known and the security administrator will make correct conclusions.

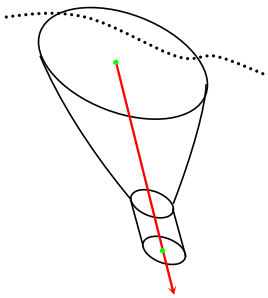

Figure 17. The fixed fitness function

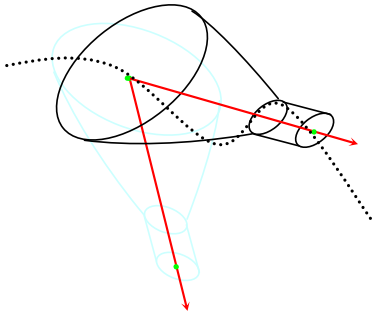

Figure 18. An adjustable FUNNEL method

The visualization of the FUNNEL method results is considered in Fig. 17. Let the solution to the problem, dotted line in fig. 17, has a 'strong inertia force' thus leaving the desired area. The interpretation shows that the fitness function in this example should be shifted as shown in fig. 18, and then the solutions will go the desired direction. We use only a set of static pictures but it is obvious that multimedia and visualization of dynamic processes will make greater effect. The hope is to realize it in ongoing projects.

D. Interactions

Briefly, the synthetic SMM control means that the overall result is defined pretty much 'by the design', by interactions between the methods than by the outcomes from each method itself.

Method interactions between INCONSISTENCY, CROSSWORD, FUNNEL, and CALEIDOSCOPE are discussed in this section. There exist much more methods under SMM control, say induction, juxtapositions etc. Not everything is described to the sake of clarity and brevity. In general all the methods are collaborative as shown above: FUNNEL-INCONSISTENCY-CALEIDOSCOPE in fig. 18 or CROSSWORD-CALEIDOSCOPE in fig. 15. Briefly, all of those interactions have been visually depicted in pictures above. On the other hand, all methods are competitive, and 'the fittest survives' principle means the following. As described, many processes should be executed in parallel but the computer resources are reserved for the high priority methods. The lowest priority belongs to the constantly active test on inconsistency which runs if any free resources. The highest priority belongs to user modeling, intruder modeling and expert- or user-ordered goals. Methods that brought a lot of successful results in the past gather higher priority. The security administrator may shift the set of priorities at any time.

Well known query processing, statistical inference and other knowledge discovery technologies will easily collaborate with the presented methods but they are included under the same SMM control. As stated above, our goal isn't a method substituting the best contemporary methods but making a good addition to them. The wide part of the funnel in fig. 18 shows that the resolution process may start using statistics in the lack of knowledge and then go to the desired direction when statistical methods are shifted by other knowledge acquisition methods. This important part of SMM is described in [13].

4. Realizations

The presented system source codes are written in different languages: C++, VB, and Prolog. It is convenient to use the applications in freeware like RDF, OWL, Ontoclean or Protégé. Many of the described procedures rely on the usage of different models/ ontologies in addition to the domain knowledge thus the latter are metaknowledge forms. In knowledge-poor environment the human-machine interactions have a great role, and the metaknowledge helps make the dialog more effective and less boring to the human. The dialog forms are divided in 5 categories from 1='informative' to 5='silent' system. Knowledge and metaknowledge fusion is always documented: where the knowledge comes from, etc. This is the main presented principle: every part of knowledge is useful and if the system is well organized, it will help us resolve some difficult situations.

We rely on nonsymmetrical reply 'surprise and win', on the usage of unknown codes in combination with well known methods, and on the high speed of automatic reply in some simple cases e.g. to halt the network connection when the attack is detected. If any part of ISS is infected or changed aiming at reverse engineering or other goals, then the system will automatically erase itself and in some evident cracking cases a harmful reply will follow. The above represented models of users and environment are used in the case. Therefore different SMM realizations are not named IDS but ISS because they include some limited automatic reply to illegal activities.

The success of the presented applications is hidden in a rather simple realization of the presented methods. We tried to make complex applications using reasoning by analogy, machine learning or statistical data mining methods but in this case the complexity of SMM is greater than NP-hard.

5. Conclusions and Future Work

The main conclusion is that all ontologies make ISS more flexible, especially dynamic ones. In the beginning almost all ontologies are poor modeled, incomplete or even partially incorrect. To improve they should evolve, and we found that conflicts or contradictions are best driving factors of such an evolution. Hence the resolution of

contradictions is one of driving factors to the usage of dynamic ontologies. The detection or resolution process uses knowledge or metaknowledge concerning evolving ontologies. Different forms of resolutions have been considered. Say, defeasible inference using exclusions allows the system to transfer the sides of conflict or contradictions to different possible worlds.

Cases have been considered where no other solution exists but only using dynamic ontologies. It is derived that many processes concerning human-machine creation are ontology-based. Our additional purpose is to show that when the machine helps to resolve the problem using its strongest features then it uses the formal, mechanical part of the research, while the heuristic part, emotions or notions like simple, beautiful, interesting should be left to the decision maker. Such human-centered methods are much more effective and less complex than automatic heuristic methods. One of the considered methods, CALEIDOSCOPE may be considered a far analogy to human-computer brainstorming methods.

To make an advanced system, we should define and use many labor-consuming models and/or ontologies. In perspective we hope that the usage of machine learning or other knowledge acquisition methods will help to construct ontologies automatically. In parallel we use the considered methods in information security projects [16].

Bibliography

[1] M. Miller. Absolute PC Security and Privacy. SYBEX Inc., CA, 2002.

[2] D. Song, M. Heywood, A. Zincir-Heywood**.** Training Genetic Programming on Half a Million Patterns: An Example From Anomaly Detection, IEEE Trans./Evolutionary Computation, no. 3, pp. 225-239, 2005.

[3] H. Kyburg*, Probability and Inductive Logic*, Progress, Moscow, 1978.

[4] The Handbook of Data Mining, N. Ye (Ed.), Lawrence Erlbaum Associates, NJ, 2003.

[5] S. Denchev and D. Hristozov. Uncertainty, Complexity and Information: Analysis and Development in Fuzzy Information Environment. Zahari Stoyanov, Sofia, 2004.

[6] G. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Upper Saddle River, Prentice Hall, NJ, 1997.

[7] V. Jotsov. "Knowledge discovery and data mining in number theory: some models and proofs," Proc. Methods and Algorithms for Distributed Information Systems Design. Institute for Information Transmission Problems of RAS, Moscow, pp.197-218, 1997.

[8] V. Zgurev and V. Jotsov, "An approach for resolving contradictions," *J. Controlling Systems and Machines* Vol. **7-8** , pp. 48-59, 1992.

[9] A. Arruda, "A survey on paraconsistent logic," *in Math. Logic in Latin America*, A. Arruda, C. Chiaqui, N. Da Costa, Eds. North-Holland, Berlin NY, pp. 1-41, 1982.

[10] Essential RDF features: www.w3schools.com/rdf

[11] Introduction to OWL: www.w3.org/TR/owl-ref/

[12] Jotsov V., Semantic Conflict Resolution Using Ontologies, Proc. 2nd Intl. Conference on System Analysis and Information Technologies, SAIT 2007, RAS, Obninsk, September 11-14, 2007, vol. 1, pp. 83-88.

[13] V. Jotsov. "Evolutionary parallels," *Proc. First Int. IEEE Symp. 'Intelligent Systems'*, T. Samad and V. Sgurev (Eds.), Varna, Bulgaria, vol. **1**, pp. 194-201, 2002.

[14] V. Jotsov. "Knowledge acquisition during the integer models investigation," *Proc. XXXV Int.Conf. "Communication, Electronic and Computer Systems"*, Tecnical University of Sofia, pp. 125-130, 2000.

[15] A. Goel, "Design, analogy and creativity," *IEEE Expert/Intelligent Systems and Their Applications*, vol. 12, no. 3, May 1997.

[16] V. Jotsov, V. Sgurev. "An investigation on software defence methods against an illegal copying," *Proc. IV Int. Sci. Conf. 'Internet - an environment for new technologies'*, vol. **7**, V. Tarnovo University 'St. St. Kiril and Metodius', pp. 11-16, 2001.

Author's Information

Vladimir S. Jotsov (В.С. Йоцов): e-mail i@AAAaaa.biz

Intsitute of Information Technologies of the Bulgarian Academy of Sciences;

State Institute of Library Studies and Information Technologies; P.O.Box 161, Sofia 1113, BULGARIA;