ONTOLOGY-DRIVEN INTRUSION DETECTION SYSTEMS

Vladimir Jotsov

Abstract*: We consider and analyze different types of ontologies and knowledge or mataknowledge connected to them aiming at realization in contemporary information security systems (ISS) and especially the case of intrusion detection systems (IDS). Human-centered methods INCONSISTENCY, FUNNEL, CALEIDOSCOPE and CROSSWORD are algorithmic or data-driven methods based on ontologies and interacting on a competitive principle. They are controlled by a synthetic metamethod SMM. It is shown that the data analysis in the field frequently needs an act of creation especially if it is applied in a knowledge-poor environment. It is shown that human-centered methodsare very suitable for resolution of the quoted tasks.*

Introduction

Contemporary information security systems (ISS) and especially those Internet-based are primarily based on the usage of intelligent methods. The case of intrusion detection systems (IDS) is machine learning oriented, and some of them are using data mining [1,2]. Such sophisticated technologies are time- and labor-consuming, it is very hard making them satisfy the demands for results convergence and low computational complexity. However designers and customers accept such difficulties trying to gain from higher security of such decision support applications. Here the base concept is to make a powerful human-centered system combined with firewalls or other passive security tools which complex defends against different groups of intruders. It is obvious 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 ontologies usage in different decision support methods and applications in data mining, web mining and/or data warehousing.

Usually ontologies are issued to support methods/applications to probabilistic, fuzzy inference, or uncertainty processing [3-6]. Our research shows other [7], sometimes nonstandard ways that are not defeating the other contemporary research but are making something in addition to well known methods, and so are useful to be combined. 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 it [8]. This is found to be the best way to self-improvement via the correction of incompleteness or inconsistency. On contrary to other machine learning methods, our proposal is ontology-driven and isn't heuristic by nature. 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. They 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 full of heuristics, which restricts its future development. Instead we offer making the machine the best human's advisor which 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 filed: human-machine creation.

Application results are considered in Section 4. The above quoted research have been never realized 'all in one' system because of its high complexity. However we used a big variety of method combinations under the 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 (using ¬ equivalent to 'not'), etc. η is a type of negation, strong negation in case, and square brackets embrace different words used to represent explicit strong negations in text.

$$\{\eta\}\ [\text{no, not, не, нет}]. \tag{1}$$

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).

$$\{U\}\ [\eta\text{: } A, B]. \tag{2}$$

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 ontology (3).

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

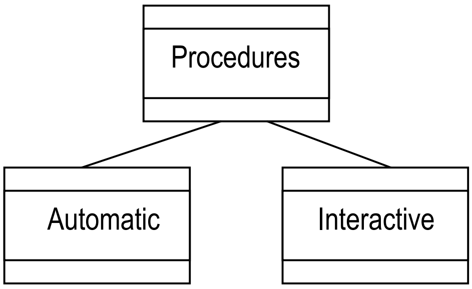


Figure 1. Ontology for a syntactic contradiction

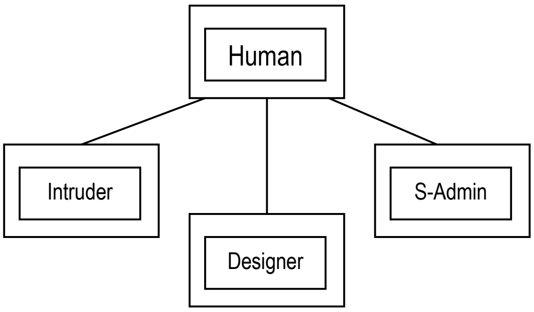


Figure 2. Ontology for a semantic contradiction

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: χ_1, χ_2, …χ_z.

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

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.

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 the essential part of both conflicts or contradictions from (2) and (3) isn't the ontology knowledge itself but the *metaknowledge* controlling 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.

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 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.

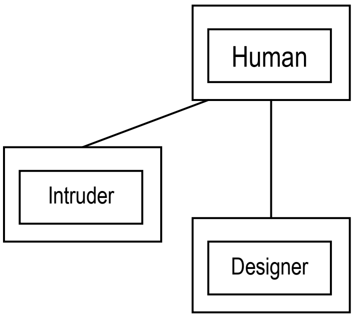


Figure 3. Ontology for conflict situations

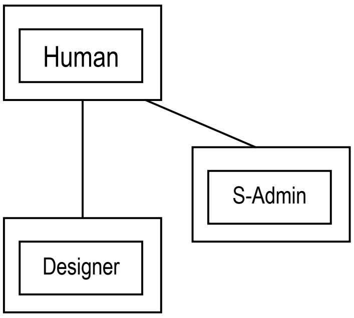


Figure 4. Ontology for contradiction situations

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 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 ontology knowledge, and using only static situations. We need the dynamic picture to decide if we have no conflict or knowledge conflict or the ultimate form, contradiction. On the other side, when the situation dynamics is investigated, pretty often we turn to ontology corrections due to its incompleteness or incorrectness. In this situation the main conclusion for us is the following. We need 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 effective applications of ideas from nonclassical logics and they are one of base parts since may decades 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; in other situations 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. As a result our contradiction resolution methods have been upgraded to a machine learning method i.e. learning without teacher which is rather effective in case of IDS.

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, or other heuristic methods. Those methods show low learning rate and high design costs. On contrary we offer machine self-improvement via contradiction or knowledge conflict resolution. The knowledge base is improving after every such resolution process. 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 [12] and it gives better results if the intermediate solutions, *hypotheses* are tested in different models [13]. 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 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 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 theoretical 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 [11, 12]. Below we choose the formal description that is complemented with explanations in an analogous manner as the way to reduce the 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. The 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 like the place of KDS'1993 – Interhotel

Sandanski... The more the crosspoints are, the easier is the solution of the horizontal sentence. The approach for the CROSSWORD is *even easier*. Here both the easy meanings and the difficult ones are from one domain, therefore there exists an additional help to find the final solution.

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, and G_1 is explored in the area defined by the constraints V_1-V_2-V_3 in fig. 5.

The constraints V_j are not necessarily linear. Nonlinear V_j are depicted in fig. 5. 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 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. 5 and the goal resolution complexity falls significantly.

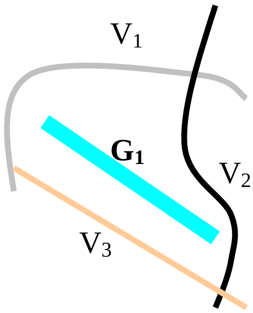


Figure 5. Example of nonlinear constraints

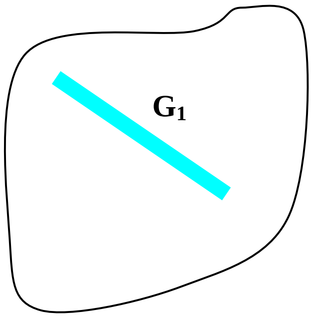


Figure 6. The goal is inside an ontology

Another situation reducing the resolution process is depicted in fig. 6 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. 6 is the proof *on contrary* when it is impossible the goal to be outside the considered ontology. Comparisons between two examples from fig. 5 and fig. 6 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.

$$K_i \in K,\ i=1,2,\ldots n:\ G_1 \cap K_i \neq \varnothing;$$

$$L_j \in L,\ j=1,2,\ldots m:\ G_1 \cap L_j = \varnothing;$$

$$S=(G_1 \cap K_1), T=(G_1 \cap K_n); S \neq T; x_1, y_1, z_1 \in S;\ x_2, y_2, z_2 \in T; \tag{5}$$

$$\frac{x-x_1}{x_2-x_1}=\frac{y-y_1}{y_2-y_1}=\frac{z-z_1}{z_2-z_1}$$

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 (5) is replaced by K_i from the previous iteration and so on.

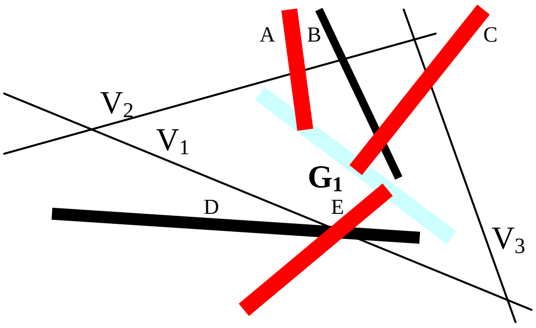


Figure 7. The CROSSWORD method goals and constraints

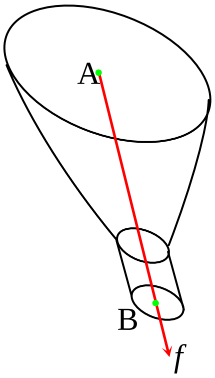


Figure 8. The FUNNEL method

Fig. 7 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. 7. The search restriction is done by V_1, V_2, and V_3 as considered above in fig. 5. 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.

B. The 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 [12], 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.

$$A\frac{d^n x}{d^n t}+B\frac{d^n y}{d^n t}+C\frac{d^n z}{d^n t}\le D \qquad (6)$$

$$Ex+Fy+Gz\le H \qquad (7)$$

where (6) is a system of non-linear constraints and (7) 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 (6) and (7).

$$f^*(t_0) = f(t_0) + \sum_i k_i f_i(t_0) \tag{8}$$

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 *inertion 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:

$$k_i(t_0) = \frac{k_i^0}{1 + D_0 - D} \tag{9}$$

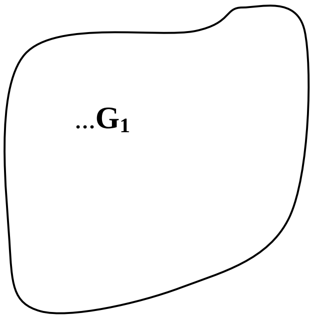
where k_i^0 are the initial meanings coinciding with k_i from (8) and (t_0) are the respective coefficients in point t_0, D is the initial point in the investigated domain–a beginning of the solution and D_0 is an orthogonal projection of t_0 upon the straight line L parallel to *f* where $D \in L$. 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. 8.

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. 8, 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. The 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. Many of the discribed methods already 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.



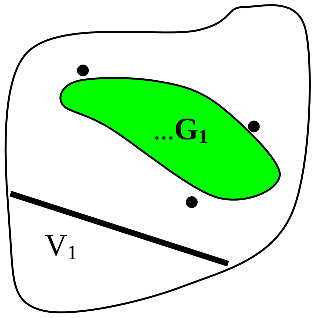
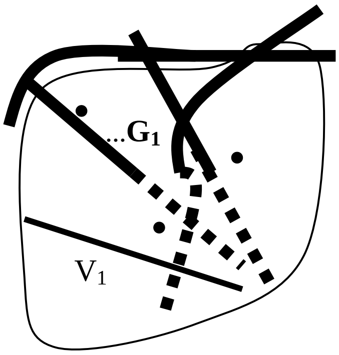


Figure 9. CROSSWORD -1: location of the goal

Figure 10. CROSSWORD -2. Two different constraint types

Fig 9 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. 10 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.



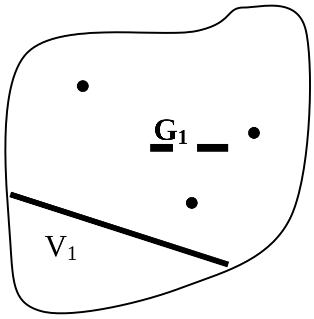


Figure 11. CROSSWORD -3

Figure 12. CROSSWORD -4

Fig. 11 shows same constraints and three resolution constraints making intersections with the desired goal G1. A part of other constraints helping define the three resolution constraints is depicted. In Fig. 12 is shown that the two right intersection parts are joined, and the left part is enlarged using knowledge modelling, binding and logic methods. Thus a big part of the goal is known and the security administrator will make correct conclusions.

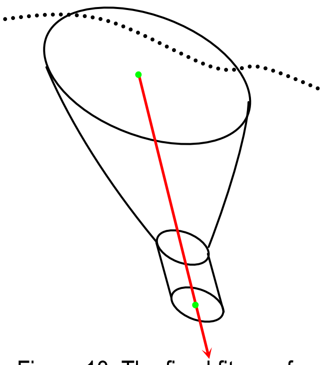


Figure 13. The fixed fitness function

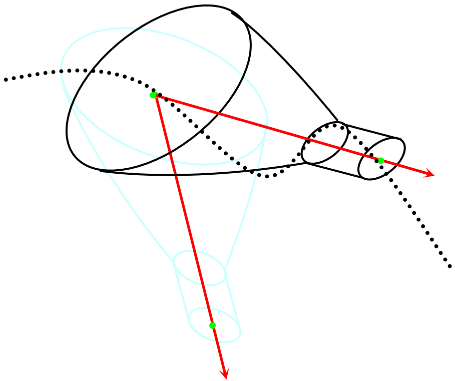


Figure 14. The adjustable FUNNEL method

The visualization of the FUNNEL method results is considered in Fig. 13. Let the solution to the problem, dotted line in fig. 13, has a 'strong inertia' thus leaving the desired area. The interpretation shows that the fitness function in this example should be shifted as shown in fig. 14, 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 control by SMM means that the overall result is defined 'by the design', by interactions between the methods much more than by the results given by each method itself.

Four methods are discussed in this section, INCONSISTENCY, CROSSWORD, FUNNEL, and CALEIDOSCOPE. There are much more methods under the SMM control: induction, juxtapositions etc. Not everything touching the problem is described to the sake of clarity and brevity. In general all the methods are collaborative as shown above: FUNNEL-INCONSISTENCY-CALEIDOSCOPE in fig. 14 or CROSSWORD-CALEIDOSCOPE in fig. 11… On the other hand, all the methods are competitive where '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 is the constant inconsistence test, it runs if some free resources. The highest priority are user modeling, intruder modeling and expert- or user-ordered goals. Methods that brought many successful results in the past get higher priority. The security administrator may change the set of priorities.

Query processing, statistical inference and other knowledge discovery technologies may collaborate with the presented methods but they are included under the same SMM control. As stated above, our goal isn't to make a method that will substitute the best contemporary methods but SMM is a good addition to them. The wide part of the funnel in fig. 14 shows that the resolution process may start using statistics in the lack of knowledge and then go the desired direction when statistical methods are shifted by other knowledge acquisition methods. This important part of SMM is described in [12].

4. Realisations

The presented system source codes are written in different languages: C++, VB, OWL and Prolog. 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 our main principle: every knowledge is useful and if the system is well organized, it will help us resolve some difficult situations.

We rely on nonsymmetric 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 here is that many processes concerning human-machine creation are onology-based. Our additional purpose is to show that when the machine helps to resolve the problem using its strongest features/formal part and the heuristic part/emotions/notions like simple, beautiful, interesting are left to the decision maker, then the human-centered methods are rather effective and simple.

There exists no free cheese… To make an advanced system we should define and use many labor-consuming ontologies. In perspective we hope use machine-learning or other knowledge acquisition methods to construct ontologies. In parallel we use the considered methods in older projects [14].

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] D. E. Goldberg, The Design of Innovation Lessons from and for Competent Genetic Algorithms, Kluwer, NY etc., 2002

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

[12] 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.

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

[14] 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

V.S.Jotsov (В.С. Йоцов): State Institute of Library Studies and Information Technologies;

Intsitute of Information Technologies of the Bulgarian Academy of Sciences.

P.O.Box 161, Sofia 1113, Bulgaria