Adjustable Fusion to Support Cyber Security Operators

François-Xavier Aguessy, Olivier Bettan, Romuald Dobigny, Claire Laudy, Gaëlle Lortal, David Faure

Adjustable Fusion to Support Cyber Security Operators

Aguessy François-Xavier[1], Bettan Olivier[1], Dobigny Romuald[2], Laudy Claire[2], Lortal Gaëlle[2], Faure David[2]

[1] Cyber Security Lab, SiX/Theresis, Thales Solutions de Securité & Services, Campus Polytechnique, 1 avenue Augustin Fresnel, 91767 Palaiseau cedex, France
[2] Analysis and Reasoning in Complex Systems Lab., Thales Research & Technology, Campus Polytechnique, 1 avenue Augustin Fresnel, 91767 Palaiseau cedex, France

`name.surname@thalesgroup.com`

Abstract. Cyber security operators use Security Information and Event Management systems to process and summarize the huge amount of heterogeneous logs and alerts. However, these systems do not give to the operator a concise view of the attack status or context, a mandatory feature to understand and remediate properly a threat. Moreover, the number of alerts to analyze for a single information system is high, and thus requires to be split into several levels of responsibility distributed among several operators. This layered security monitoring implies a decision problem as well as an automation problem tackled in this paper with the support of an attack graph-based feature. An attack graph is a risk assessment model that accurately describes, in a concise way, the threats on an information system. In this article, we describe how an attack graph can be used for pattern searching and fusion algorithms, in order to add context to the alerts. We also present recommendations for designing future interactive application based on adjustable fusion and a risk assessment model, for cyber security monitoring.

1   Introduction

As information systems are getting ever more complex, they produce large amount of heterogeneous logs and alerts that operators working in Information Security Operations Center (SOC) cannot process without smart aggregation. They thus need to use Security Information and Event Management (SIEM) systems that collect, aggregate, normalize, correlate and report the events generated in an information system. SIEM solutions provide a dynamic view of the security events in a system, but they do not give to operators a concise view of the attack status and context, with knowledge of possible future, which may be necessary to understand and remediate properly a cyber security threat.

Security operators have a huge number of alerts to deal with for a single information system and thus this process has been split into several levels of responsibility shared among several operators. We are facing here a collaborative decision problem and not solely an automation problem. In these approaches at use in SOCs, responsi-

bility involvement, collaboration and awareness of each stakeholder is crucial to ensure the quality of a shared decision making.

Some risk assessment models such as attack graphs allow an operator to understand the threats targeting an information system. An attack graph is a risk analysis model regrouping all the paths an attacker may follow in an information system. It is composed of nodes, representing the hosts that can be exploited by an attacker. Nodes are linked together with edges, representing the attacks that can be done between these hosts. Contrary to the outputs of SIEM (alerts on ongoing attacks), these threat models represent attacks that are likely to happen, with their context of occurrence. However, it gives a static view of the system with no easy way to know which part of the information system is currently under attack. There is thus a strong need to combine both approaches, in order to take advantages of dynamicity and correctness out of SIEM systems, and of contextual and concise view out of risk assessment models. As the risk assessment models are based on big graphs, and the SIEM events are in some ways a pattern of such graphs, graph fusion and query algorithms can be used to find the patterns that have been detected in the whole graph. We aim at studying in this article, recommendations for designing future interactive application based on adjustable fusion and a risk assessment model, for cyber security monitoring.

This paper is organized as follows: in Section 2, we describe the state of the art of the technologies involved: SIEM, responsibility transfer, graph-based risk assessment models and pattern-matching and fusion algorithms. Section 3 presents the graph model used for pattern matching. Section 4 integrates the human aspects of this decision problem, by introducing graph fusion. Section 5 concludes and put this work into perspective.

2 State of the art

2.1 Security Information and Event Management (SIEM) systems

Several correlation methods can be used in Security Information and Event Management systems, to reduce the number of alerts and correlate them. They can be regrouped in 3 categories [1]:

- **similarity-based methods** [2,3]: aggregating the alerts by using their similarities on attributes or time,
- **sequential-based methods** [4,5]: alerts are regrouped, according to a model (pre/post conditions),
- **case-based methods** [6,7]: research of specific patterns following scenarios defined in expert rules.

Products using alert correlation are generally one main component of SIEM, which also collect and report security events, related to these alerts. Most of these tools (open source [8,9] or commercial [10,11,12,13]), often used by SOCs operators, first reduce and cluster the alerts, then implement a rule-based correlation approach, using an expert rules correlation engine. This is a quite simple but efficient process. How-

ever, detection capabilities of pattern-based SIEM tools are optimal for known and well characterized threats.

2.2 Responsibility transfer and collaborative work

SIEM approaches at works in SOCs propose a segmentation of threats management by human operators into 3 levels: (1) security tickets creation due to alerts raised by security devices and first qualification; (2) Assessment of security incident resulting from alerts correlation and possible remediation proposal; (3) Incident investigation and technical escalation [14].

Figure 1 represents the common escalation process and responsibility transfer in a SOC, which follows the start of an incident (generally with an alert issued by the SIEM), until the incident is closed.
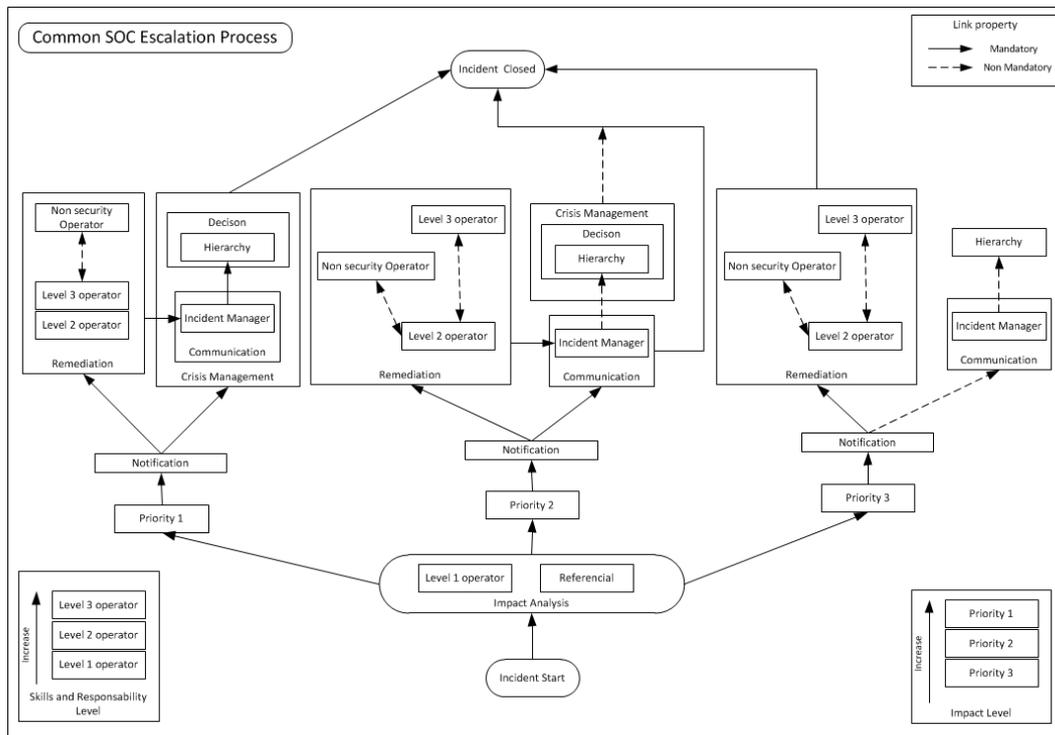

Fig. 1. Common SOC Escalation Process

2.3 Graph-based models for cyber security

Attack trees are a well-known multi-step attack model which is very important since it is one of the first graphical models to have been proposed for security assessment.

The notion of attack tree was introduced by Bruce Schneier [15]. In his article, the attack trees are defined informally as a tree with AND/OR nodes which describe exactly the possible or required steps to do an attack and arcs modeling dependencies between these steps. The goal of the attack is located in the root of the tree and the basic actions used to achieve this goal are leaf nodes. Attack trees were formalized by Mauw *et al.* [16].

The main limitation of attack trees is that they only describe one main attack. To respond this limitation, attack graphs have been created. An attack graph is a model that regroups all the steps that an attacker may follow in an information system during an attack from both outside or inside the local network. It has been first introduced by Phillips and Swiler [17]. This formalism has been widely used ever since, thus many heterogeneous models are now behind the name attack graph. Generally, vertices (also called nodes in the literature) represent opportunities in an information system or actions that can be done by an attacker, and edges (also called arcs in the literature) represent the dependency relations between the opportunities and/or actions. An attack graph can be built using information about the potential exploits that can be carried out on a network or using existing vulnerabilities databases. A summary of the state of the art on the early papers about attack graphs (from 2002 to 2005) has been done by Lippmann and Ingols in [18]; a more recent by Kordy *et al.* in [19].

Attack graphs can be regrouped into two main categories: logical attack graphs and topological attack graphs. A logical attack graph is an AND/OR direct graph in which nodes are logical facts containing what is reachable by an attacker. This is for example the case of the graphs generated with the attack graph engine MulVAL [20]. A topological attack graph may be generated from a logical attack graph, but it gives a more concise view of the possible attacks, as vertices model machines or IP addresses, and edges represent an attack step: the way to exploit a host from another host. For example, the attack graph engine Netspa [21] generates such topological attack graphs. The main advantage of the topological attack graphs rather the logical ones, is that this graph is much more concise and easy to understand for a security operator, as it follows the usual view of the information system.

Attack trees and attack graphs are very close models representing multi-step attacks with respectively one or several attack goals. They can contain very accurate description of attacks (with logical attack graphs/trees) or more high-level vision of attacks (with topological attack graphs/trees). However, these models were not built to represent on-going attacks. They only contain the description of the potential attacks that could happen in an information system.

2.4   Pattern matching in graph and fusion

High-level information fusion. Soft data fusion is an ever growing trend in the information fusion community. More and more tracks dedicated to soft information are organized within the International Conference on Information Fusion over the years and numerous authors stress the need for soft data management and fusion. For instance, in the detection of people and complex activities, the use of soft information sources is critical. The authors of [22] describe a 5 year program involving several

academic actors that aim at addressing major stakes of soft data fusion. It includes the development of a framework as well as evaluation methods.

In addition, many authors relate about new issues raised by soft data fusion. Among them [23] and [24] quote natural language processing, transformation of data into comprehensive and semantic data structures, soft data association and graph matching. As the automation of soft data fusion is a very challenging issue, due to, e.g., error estimation, normalization and context extraction for information interpretation, the authors propose a mixed approach that embeds the participation of a human analyst to the fusion process.

We previously developed a framework based on graph structures, graph algorithm and similarity measures for soft data fusion managing inconsistencies (InSyTo Synthesis v1 [25,26]). The use case presented in [27] dealt with the management of an information network containing descriptions of entities (companies, universities...) that collaborate through several media such as research project, scientific papers and so on. The InSyTo Synthesis framework was used to enable non-redundant information addition to the information network, as well as graph based information query.

Information query in graphs. The graph fusion algorithm relies on the search for matches between sub-graphs and more precisely for a maximal matching subgraph. The idea is to find the largest subgraph of a first graph that cannot be distinguished from a subgraph of second graph. Maximal subgraph matching is used in order to determine, where to add information in an information graph, and which parts of two information graphs are redundant and should thus be fused rather than be repeated twice in the graph resulting from the fusion.

The decision problem of whether two graphs match is well studied and is in NP [28]. The problem of finding matching subgraphs, known as the subgraph isomorphism problem, is known to be NP-complete and difficult to solve in parallel. This problem is well studied and have led to a lot of algorithms, among which several have been parallelized [29,30]. The InSyTo Synthesis fusion algorithm relies on subgraph isomorphism and maximum subgraph isomorphism algorithms. Both those algorithms must solve highly combinatory problems, on which the execution time may become too long to satisfy user requirements. To manage this problem, the InSyTo Synthesis subgraph isomorphism algorithm is partially parallelized.

3   Attack model and alerts pattern matching

The goal of the process we describe in this article is to help security operators to identify the attacks that are currently happening in a model of their information system, while taking into account human computer interaction, in order to improve the decision process in Security Information and Event Management systems.

As seen in the state of the art, on one hand, some SIEM tools gather and regroup alerts, but are not sufficient to give the context needed by an operator to properly understand and treat them. On the other hand, some models of the information system represent the attacks that can happen, but they give a static view of the system with no easy way to know which part of the information system is currently under attack.

Such models are represented by a graph of the information system, in which fusion and query algorithms can be used.

3.1 Information representation of the attack model

Topological Attack Graph. The model on which we will base our pattern matching and fusion processes is a topological attack graph, which describes all the attacks that are possible in an information system. This choice has been led by the ability of such a model to contain an accurate description of the attacks, while keeping a concise view, easy to understand for a security operator, as it is based on the topological representation of an information system. This graph is constituted of vertices, representing hosts, and edges, representing potential attacks. Each edge is associated with a set of metadata describing, for example, which vulnerability is exploited on this path, the probes that are related to this path, and the log message that may be provided, if such path is actually followed.

Alerts. The alerts that are issued by security probes (such as SIEM, Host or Network Intrusion Detection System (IDS), firewalls…) are parsed, in order to extract meaning from the log message (probe, source host, destination host, vulnerability exploited…). Such alerts can be units (e.g. IDS alerts) or multiples (e.g. SIEM logs that describe multi-steps attack). A security operator may receive numerous alerts in a small time interval. Unit alerts that are received in a short interval or multiple alerts already correlated are part of one main attack scenario and constitutes an attack pattern. The security operator wants to match this pattern in the whole graph, to know the reals attacks that are currently happening, and what can happen next.

A graph based information representation. We use Basic Conceptual Graphs [31] to represent information. Basic Conceptual Graphs are bipartite graphs containing concept and relation nodes. Figure 2 gives an example of a conceptual graph. The rectangular boxes represent concept nodes and the ovals represent relation nodes.

Graphical form

Linear form

vulnerableTo([Host: db@db], [Vulnerability: CVE-1234 @CVE-1234]),
canAttack([Host:webServer@webServer], [Host: db@db]).

Fig. 2. Conceptual graph example

The concepts represent the *things* or entities that exist. For instance, in Figure 2, the concept `Host:webserver` represents an instance of a `Host` object whose name is `webserver`. The relation nodes indicate the relations that hold between the different concepts of a situation.

The types of concepts are organized into a hierarchy. Therefore, a specialization/generalization relation may be defined between several graphs [26]. This relation is used within the query function. Answers to a query graph are sub-graphs of the data graph that are more specific than the query graph. This would allow to use this approach with hierarchical attack graphs.

3.2 Graph-based Information Query

The query algorithm relies on a generic sub graph matching algorithm, which itself uses specific fusion strategies [25]. The graph matching component is in charge of the structural consistency between the query and information graphs. The fusion strategy part is made of compatibility functions over elements of the graphs. They enable the customization of the generic algorithm according to the context in which it is used. Within the query function, we use a *subsomption* strategy and a whole-structure conservation mode.

Query algorithm. The inputs of the query function are two graphs. The information graph is a big network (the topological attack graph), while the query graph is a relatively small one (the alerts pattern). Our approach to parallelize the graph matching process is to split up the subgraph matching process. A first phase manages the node to node comparisons of the query and data graphs. A second step is in charge of preserving the structure of the graphs. This process has been described more deeply in [25].

Graph structure preservation. Once the candidate answers to each subgraph of the query are processed, all the combinations of answer subgraphs are provided, preserving the original structure of the data graph. The candidate answer subgraphs are assembled one with another, if and only if their association respects the structural constraints of the initial data graph. The connectivity between the relations through the concept nodes is checked.

Queries in topological attack graphs. This query algorithm applies to finding alert patterns in topological attack graphs stored in Basic Conceptual Graphs. The topological attack graph stands in for the information graph, as it is the big graph, in which all the knowledge of the information system (topology, vulnerabilities, possible attacks…) is stored. The alert patterns stand in for the query graph, as they represent a smaller graph, subgraph of the information graph. This algorithm outputs the position in the topological attack graph of the detected patterns.

4 Human Computer Interaction in SIEM

4.1 Human Computer Interaction drawbacks

SOC and SIEM technologies have several drawbacks in terms of HCI. The 1st level (security ticket creation) is tedious and repetitive work, which doesn't enable global situation awareness; The 2nd level (assessment of security incident) generates alert transmission on the different levels of SOC slowing down the logs analysis processes; As for the 3rd level (incident investigation and technical escalation), the added value from cyber security operators is at the crossing of the three levels and in the incident context conceptualization. Visual analytics techniques supporting cyber security monitoring suffer from lack of interactivity and visualization overload despite recent works [32] and are often not structured advisedly. The approach proposed here will fuse the information and then present less information to the operators but more synthetized thanks to the fusion process. This will enable the operators to better understand the situation and take a better decision even facing large/complex attacks.

Moreover, to support the operators in their analyses and decisions, we propose to study responsibility sharing and awareness between the operators at different level [33] to propose design recommendations for a future interactive application based on adjustable fusion and increasing the collaboration between the operators to increase their global understanding and efficiency.

4.2 Operator Dependent Fusion Strategies

The InSyTo Synthesis platform encompasses a generic graph based fusion algorithm that is used for the three functions (information fusion, information synthesis and information query). The usage of the algorithm (parameters and launch mode) determines the function that is realized. The fusion algorithm is made of two interrelated components (see Figure 3). The first component is a generic subgraph matching algorithm, which itself relies on the use of fusion strategies. The graph matching component takes care of the overall structures of the initial and fused observations. It is in charge of the structural consistency of the fused information, regarding the structures of the initial observations, within the fusion process.
The fusion strategy part is made of similarity, compatibility and functions over elements of the graphs to be fused. They enable the customization of the generic fusion algorithm according to the context in which it is used. The context encompasses the application domain, the semantics of the information items and user preferences.

The fusion strategies enable to manage the discrepancies that may be observed in observations of the same situation by different sources.
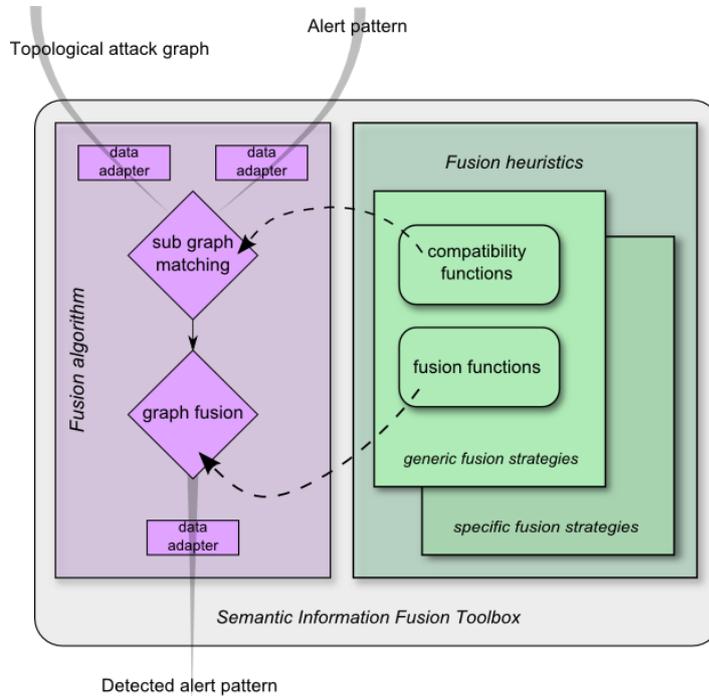
Fig. 3. Fusion architecture

5   Conclusion and Perspectives

We presented in this article the status of current approaches at use in Information Security Operations Center, where security operators based their analysis on results of Security Information and Event Management systems. As these operators have a huge number of alerts to treat, this process has been split into several levels of responsibility distributed among several operators, which cause a collaborative decision problem. To improve the decision, we proposed a process using pattern matching and information fusion algorithms, reasoning on topological attack graphs. This process, allowed taking advantages of dynamicity and correctness out of SIEM systems, and of contextual and concise view out of risk assessment models. The adjustable fusion process could be refined to propose solutions enabling to avoid bystander effect, social loafing or group think in collaborative decision making [34]. We assume that to structure the types of interaction with the system(s) following the Sheridan and Verplank model [35] would enable operators to adapt the correlation process to their situation awareness and ground truth modeling needs.

6 References

1. Salah, S., Maciá-Fernández, G., & Díaz-Verdejo, J. E. (2013). A model-based survey of alert correlation techniques. Computer Networks, 57(5), 1289-1317.
2. Zhuang, X., Xiao, D., Liu, X., & Zhang, Y. (2008, December). Applying data fusion in collaborative alerts correlation. In Computer Science and Computational Technology, 2008. ISCSCT'08. International Symposium on (Vol. 2, pp. 124-127). IEEE.
3. Ahmadinejad, S. H., & Jalili, S. (2009, November). Alert correlation using correlation probability estimation and time windows. In Computer Technology and Development, 2009. ICCTD'09. International Conference on (Vol. 2, pp. 170-175). IEEE.
4. Zhaowen, L., Shan, L., & Yan, M. (2010, September). Real-time intrusion alert correlation system based on prerequisites and consequence. In Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on (pp. 1-5). IEEE.
5. Roschke, S., Cheng, F., & Meinel, C. (2011). A new alert correlation algorithm based on attack graph. In Computational Intelligence in Security for Information Systems (pp. 58-67). Springer Berlin Heidelberg.
6. Cuppens, F., & Ortalo, R. (2000, January). LAMBDA: A language to model a database for detection of attacks. In Recent advances in intrusion detection (pp. 197-216). Springer Berlin Heidelberg.
7. Katipally, R., Gasior, W., Cui, X., & Yang, L. (2010, April). Multistage attack detection system for network administrators using data mining. In Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research (p. 51). ACM.
8. OSSIM, AlienVault, http://communities.alienvault.com/
9. OSSEC, Trend Micro, http://www.ossec.net/
10. QRadar : http://www-03.ibm.com/software/products/en/category/security-intelligence
11. ArcSight ETRM Platform, HP, http://www.hpenterprisesecurity.com
12. RSA Envision: http://www.emc.com/security/rsa-critical-incident-response-solution.htm#!solution_description
13. Splunk, http://splunk.com
14. Kelley, D. & Moritz, R. (2006, January-February). Best Practices for Building a Security Operations Center. In Information Systems Security (pp. 27-32).
15. Schneier, B. (1999). Attack trees. *Dr. Dobb's journal*, *24*(12), 21-29.
16. Mauw, S., & Oostdijk, M. (2006). Foundations of attack trees. In *Information Security and Cryptology-ICISC 2005* (pp. 186-198). Springer Berlin Heidelberg.
17. Phillips, C., & Swiler, L. P. (1998, January). A graph-based system for network-vulnerability analysis. In *Proceedings of the 1998 workshop on New security paradigms* (pp. 71-79). ACM.
18. Lippmann, R. P., & Ingols, K. W. (2005). *An annotated review of past papers on attack graphs* (No. PR-IA-1). MASSACHUSETTS INST OF TECH LEXINGTON LINCOLN LAB.
19. Kordy, B., Piètre-Cambacédès, L., & Schweitzer, P. (2014). DAG-based attack and defense modeling: Don't miss the forest for the attack trees. *Computer Science Review*, *13*, 1-38.
20. Ou, X., Govindavajhala, S., & Appel, A. W. (2005, August). MulVAL: A Logic-based Network Security Analyzer. In *USENIX security*.
21. Artz, M. L. (2002). *Netspa: A network security planning architecture* (Doctoral dissertation, Massachusetts Institute of Technology).

22. Llinas, J., Nagi, R., Hall, D., & Lavery, J. (2010, July). A multi-disciplinary university research initiative in hard and soft information fusion: Overview, research strategies and initial results. In *Information Fusion (FUSION), 2010 13th Conference on* (pp. 1-7). IEEE.

23. Gross, G. A., Nagi, R., Sambhoos, K., Schlegel, D. R., Shapiro, S. C., & Tauer, G. (2012, July). Towards hard soft data fusion: Processing architecture and implementation for the joint fusion and analysis of hard and soft intelligence data. In *Information Fusion (FUSION), 2012 15th International Conference on* (pp. 955-962). IEEE.

24. Gross, G. A., Khopkar, S., Nagi, R., & Sambhoos, K. (2013, July). Data association and graph analytical processing of hard and soft intelligence data. In *Information Fusion (FUSION), 2013 16th International Conference on* (pp. 404-411). IEEE.

25. Laudy, C. (2011). Semantic Knowledge Representations for Soft Data Fusion. INTECH Open Access Publisher.

26. Fossier, S., Laudy, C., & Pichon, F. (2013, July). Managing uncertainty in conceptual graph-based soft information fusion. In *Information Fusion (FUSION), 2013 16th International Conference on* (pp. 930-937). IEEE.

27. Laudy, C., Deparis, E., Lortal, G., & Mattioli, J. (2013, July). Multi-granular fusion for social data analysis for a decision and intelligence application. In *Information Fusion (FUSION), 2013 16th International Conference on* (pp. 1849-1855). IEEE.

28. McKay, B. D. (1981). *Practical graph isomorphism*. Department of Computer Science, Vanderbilt University.

29. Plantenga, T. (2013). Inexact subgraph isomorphism in MapReduce. *Journal of Parallel and Distributed Computing*, *73*(2), 164-175.

30. Zhao, Z., Wang, G., Butt, A. R., Khan, M., Kumar, V. A., & Marathe, M. V. (2012, May). Sahad: Subgraph analysis in massive networks using hadoop. In *Parallel & Distributed Processing Symposium (IPDPS), 2012 IEEE 26th International* (pp. 390-401). IEEE.

31. Chein, M., & Mugnier, M. L. (2008). Graph-based knowledge representation: computational foundations of conceptual graphs. Springer Science & Business Media.

32. McKenna S., Mazur D., Agutter J., and Meyer M., Design Activity Framework for Visualization Design, in Proceedings of the IEEE VIS conference, 2014, Paris.

33. Montferrat, P., Lortal, G., Faure, D., and Coppin, G., 2009, Intention de transfert de responsabilité pour le travail coopératif, In Association pour la Recherche Cognitive (ARCo'09), Rouen, France, Décembre.

34. Montferrat, P., Faure, D., andLortal, G., 2009, The 'Responsibility Cube' in Maritime Surveillance Domain, in Proceedings of COGIS – Cognitive Systems with Interactive Sensors- (SEE, IET eds.), ISBN : 2-912328-55-1, 6 pages.

35. Sheridan, T. B., and Verplank, W. (1978). Human and Computer Control of Undersea Teleoperators . Cambridge, MA: Man-Machine Systems Laboratory, Department of Mechanical Engineering, MIT