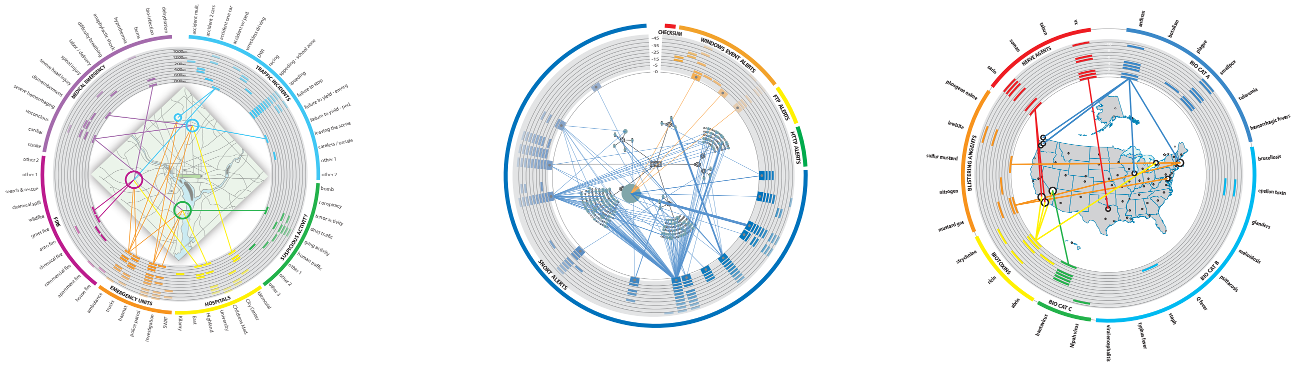
Visual Correlation for Situational Awareness

Yarden Livnat*
Scientific Computing and
Imaging Institute
University of Utah

Jim Agutter,†
College of
Architecture+Planning
University of Utah

Shaun Moon, ‡
College of
Architecture+Planning
University of Utah

Stefano Foresti§
Center for High
Performance Computing
University of Utah



(a) 911 Emergency Center

(b) Network Intrusion Detection

(c) BioWatch

Figure 1: VisAware: A novel visualization paradigm for situational awareness.

Abstract

We present a novel visual correlation paradigm for situational awareness (SA) and suggest its usage in a diverse set of applications that require a high level of SA. Our approach is based on a concise and scalable representation, which leads to a flexible visualization tool that is both clear and intuitive to use. Situational awareness is the continuous extraction of environmental information, its integration with previous knowledge to form a coherent mental picture, and the use of that picture in anticipating future events.

In this paper we build on our previous work on visualization for network intrusion detection and show how that approach can be generalized to encompass a much broader class of SA systems. We first propose a generalization that is based on what we term, the w^3*premise*, namely that each event must have have at least the *What*, *When* and *Where* attributes. We also present a second generalization, which increases flexibility and facilitates complex visual correlations. Finally, we demonstrate the generality of our approaches by applying our visualization paradigm in a collection of diverse SA areas.

CR Categories: H.5.1 [Information Interfaces and Presentations]: User Interfaces—Graphical user interfaces;

Keywords: situation awareness, network intrusion, visualization

1 Introduction

Situational Awareness (SA) is the ability to identify, process, and comprehend the critical elements of information about what is happening. The term SA comes from the world of military pilots, where achieving high levels of SA was found to be both critical and challenging [5]. The importance of SA as a foundation of decision-making and performance span many fields such as air traffic controllers, driving, power plant operations, maintenance, and military operations.

There is a growing body of research that validates the role of visualization as a means for solving complex data problems. Visualization elevates the comprehension of information by fostering rapid correlation and perceived associations. To that end, the design of the display must support the decision making process: identifying problems, characterizing them, and determining appropriate responses. It is imperative that information be presented in a manner that facilitates the user's ability to process the information and minimize any mental transformations that must be applied to the data.

In this work we focus on developing a visualization paradigm that takes advantage of human perceptive and cognitive facilities in order to enhance users' situational awareness and support decision-making. We propose a novel visual correlation paradigm for SA and suggest its usage in a diverse set of SA applications.

Recently, we proposed a new visualization paradigm for network intrusion detection (*VisAlert*) [13] as seen in Figure 1(b). The development of *VisAlert* involved traditional user-centric analysis, design and development cycles, but was focused solely on network intrusion detection. In this paper, we revisit this earlier work and examine it from a more general viewpoint, based on what we term the w^3 premise. Using this premise, *VisAlert* can be seen as a special case of a much broader class of SA systems, namely *visual cor-*

*e-mail: yarden@sci.utah.edu
†e-mail: agutterja@arch.utah.edu
‡e-mail: moonsr@arch.edu.edu
§e-mail: email@stefanoforesti.com

relation. We then propose a general framework for event correlation and provide examples of SA applications along with prototype designs for their displays. We proceed by proposing a second generalization to this framework, which introduces the notion of mixing events and resources. This final framework is both flexible and scalable, yet clear and intuitive to use. As with the first framework, we provide examples of SA application along with design prototypes of their displays.

This paper is structured as follows. Section 2 contains a brief introduction to situational awareness. Section 3 describes our earlier work on visualization for network intrusion detection and sets up the basis for its generalization. Section 4 presents two generalization frameworks and provides a collection of examples of possible application of these frameworks to various SA areas. We conclude and suggest future work in section 5.

2 Situational Awareness and Decision Making

Situational awareness was defined by Endsley [3] as *"the perception of the elements in the environment with a volume of time and space, the comprehension of their meaning, and the projection of the their status in the near future"*. The formal definition of SA breaks down into three separate levels [5]:

- Level 1 – *perception* of the elements in the environment.
- Level 2 – *comprehension* of the current situation.
- Level 3 – *projection* of future status.

The relationship between these SA levels, the mental model, and the decision-making is depicted in Figure 2.

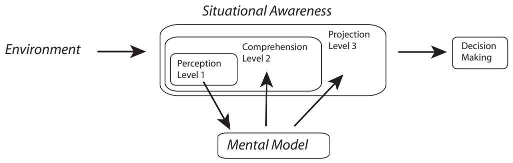


Figure 2: Situational Awareness

Level 1, perception of status, attributes and dynamics of the environment, may be gained by combination of visual, tactile, and auditory senses. Jones and Endsley [8] have shown that 76% of SA errors in pilots were related to not perceiving the needed information.

The second level in SA involves the comprehension of what the received data means in relation to the relevant goals and objectives. This may include integration of the data to form information, prioritizing and associating specific goal-related meanings and significance. In their study, Jones and Endsley [8] found that up to 19% of SA errors occured when the pilots received the necessary data (SA level 1) but were not able to correctly understand its meaning.

Level 3 SA is achieved when one can predict how the environment will be affected in the future, based on the perceived data and its meaning. One must have a good understanding of the situation and the dynamics of the system in order to achieve level 3 SA.

The perception of time and the temporal dynamics of the environment are important factors in SA. The trends developed over time can play a critical part as well. Time is a strong part of Level 2 SA and Level 3 [5].

2.1 Previous Work

The National Center for Supercomputing Applications (NCSA) has developed two applications for the detection of network incidents: VisFlowConnect and NVisionIP. VisFlowConnect uses flexible animation capabilities to display network activity. These animations can be configured to show an aggregation of network activity over time, or to more accurately reflect traffic dynamics over time. VisFlowConnect also allows analysts to filter traffic based on attributes that may help reveal network misuse. [19] NVisionIP uses a visualization structure that represents network activity at three levels, from a global view of the entire network under study down to a single machine. This scalability increases users' understanding of the relationships of part to whole and vice versa [12].

The Automated Intrusion Detection Environment (AIDE) developed by Air Force Research Lab in Rome, New York, integrates data from various intrusion detection systems into a standardized view. Using multiple data sets increases understanding of network traffic and alert intricacies [10].

An anesthesia display developed by Michels organized 32 variables by organ system, showing the absolute values of variables in relationship to a normal reference. Studies of this display showed that state changes were seen an average of three minutes sooner than in traditional displays, proving that organized structures and deviation displays can improve situational awareness [7].

Quantum 3D developed a display for the Swedish Air Force that increases ground controllers' situational awareness by generating three-dimensional views of the air space. This helped the controllers understand actual relationships between critical objects in real space. In this case, situational awareness was enhanced by remote simulations of reality more than traditional 2D radar displays [15].

3 A Visualization Paradigm for Network Intrusion Detection

Rather than describe a theoretical framework for a class of SA systems without a concrete example, we first present a real system for which we have already developed a working prototype. We then build on that work and develop the theory leading to more general frameworks in Sections 4.2 and 4.4. Finally, we provide concrete examples within these general frameworks and demonstrate possible designs of their displays. We wish to emphasize that the design process we used in the development of *VisAlert* was based on a long and methodical user-centric design cycle. The process involved a psychology team; interviews with network experts; and many analysis, design and evaluation phase cycles. However, in order to facilitate the generalization discussion which is the main goal of this work, we present *VisAlert* with respect to a different framework. This framework is closely related to the w^3 framework that along with its generalization are the main topic of the next section.

3.1 Network Intrusion Detection

The spread of malicious network activities poses great risks to the operational integrity of many corporations, institutions, and organizations, in addition to imposing heavy economic burdens. Intrusion detection systems (IDS) analyze network traffic and host-based processes, in an attempt to detect such malicious activities. The proliferation of different IDSs and the sophistication of attacks leads to a large number of alert types. This complexity is compounded by the sheer number of alerts these systems generate and a high rate of false positives.

One approach to resolving these issues is to correlate various alerts by common attributes. This approach is based on the premise that while a false positive alert should not exhibit correlation to other alerts, a sustained attack will likely raise several alerts. Furthermore, real attack activities will most likely generate multiple alerts of different types. There exists a large body of work aimed at correlating these disparate alert logs based upon clustering, probability, and similarity to predefined attacks [17, 1].

Visualization has only recently been applied to computer security data analysis, and these applications have typically been limited in their capabilities. Historically, visualization has been applied extensively to network monitoring and analysis, primarily for monitoring network health and performance [11, 6]. The few visualization techniques that have been developed for intrusion detection have been limited as to their applicability and effectiveness [16, 14, 18, 2].

3.2 VisAlert

Typically, alerts are correlated based on either their *time* or *type* attributes. However, the real value of an alert is with respect to the local node(s) it pertains to. It is, in fact, the preservation of the nodes' status and integrity that is the main focus of IDS to begin with.

We also distinguish between an alert's *definition* and an alert's *instances*. An alert definition is static and describes the preconditions and meaning of an alert. An alert instance, on the other hand, refers to a particular point in time when the alert preconditions were met. An alert instance may also include detailed information about how and what triggered a particular alert. Correlating alerts and nodes, therefore, means correlating alert instances with respect to a particular node.

A possible approach is to use a three-dimensional Cartesian coordinate system and map the *type*, *time*, and *node* to the three axes. In this configuration, a network event is represented by a three-dimensional point. There are several problems with this approach. First, the points do not exhibit any obvious correlation; for example, two nearby points may not share any attribute with each other. Second, the three-dimensional view introduces visual obstacles such as occlusion and depth perception. Finally, the visual perception will depend on the user's specific viewpoint, which only adds another unnecessary degree of freedom to an already complicated situation.

In contrast, we base our approach on representing the network alerts as connections between two domains. These two domains are a one dimensional domain representing the *node* attribute, and a two-dimensional domain representing the *time* and *type* attributes. Note that the *node* and *type* spaces are finite while the *time* space is semi-infinite. A network alert instance, in this scheme, is thus a straight line from a point in the *type-time* domain to a point in the *node* domain, as shown in Figure 3. We choose to separate the *node* attribute from the *type* and *time* as nodes provide a more or less static set of objects that we can use as visualization anchors for the transient alert instances.

We can now expand the *node* domain into a two-dimensional domain, which enables to layout the nodes in a more flexible and meaningful way for the user, such as the network topology map. The new design layout, as shown in Figure 4, maps the *node* domain onto a finite circle, while the *type-time* domain is wrapped around it in the shape of a ring. We maintain the orthogonality of the *type* and *time* spaces by mapping the *type* attribute to the angle around the circle and the *time* attribute to the radial direction from the center of the circle.

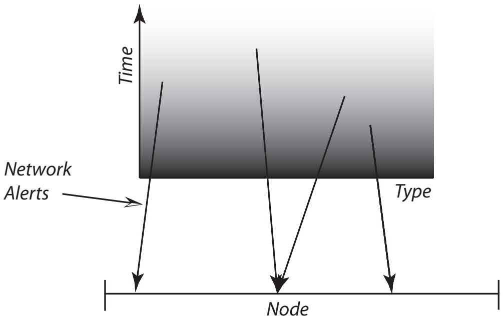


Figure 3: Domains: Representing network alerts as lines between two domains. The design facilitates the correlation of alerts with respect to the same nodes.

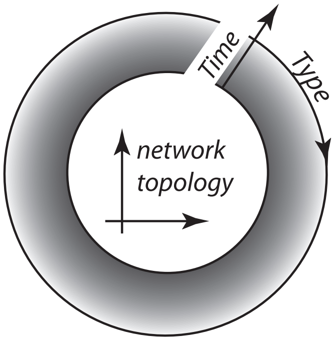
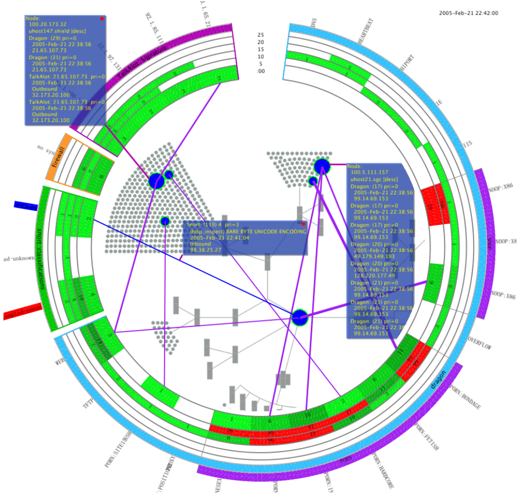


Figure 4: Mapping: The finite two-dimensional *node* domain is mapped into a central circle containing the network topology map, while the *time-type* domain is mapped to a ring around it.

Finally, to reduce the possible cluttering when showing all the alerts simultaneously, we divide the *time* space into varying intervals (rings) and show the alert instances for the most recent history period (inner most ring). For alerts that took place in the past, we display only the number of alerts that occurred during a specific time period on its corresponding ring.

3.2.1 Additional Visualization Cues

The design layout of the *VisAlert* display is shown in Figure 1(b) and includes various host-based and network-based alert logs. Figure 5 shows the display of our implementation of *VisAlert* which includes other visual indicators that encode additional information to increase the situational awareness of the user. We have adopted a method of increasing the size of nodes experiencing unique alerts. The assumption is that a resource/node on the topology that is experiencing multiple unique alerts from various host- and network-based sources has a higher probability of malicious activity than one experiencing only one alert. The size of the node is a very clear indication and is easily distinguishable from other machines in order to focus the user's attention so he or she can take action and correct the problem on the suspect machine(s).



Node:
100.20.173.32
uhost147.shield [desc]
Dragon: (29) pri=0
2005-Feb-21 22:38:56
21.65.107.73
Dragon: (31) pri=0
2005-Feb-21 22:38:56
21.65.107.73
TalkAlot: 21.65.107.73 pri=0
2005-Feb-21 22:38:56
Outbound
32.173.20.100
TalkAlot: 21.65.107.73 pri=0
2005-Feb-21 22:38:56
Outbound
32.173.20.100

Snort: (119) 4 pri=3
(http_inspect) BARE BYTE UNICODE ENCODING
2005-Feb-21 22:41:04
Inbound
98.38.25.27

Node:
100.5.111.157
uhost21.sgc [desc]
Dragon: (17) pri=0
2005-Feb-21 22:38:56
99.14.69.153
Dragon: (17) pri=0
2005-Feb-21 22:38:56
99.14.69.153
Dragon: (17) pri=0
2005-Feb-21 22:38:56
99.14.69.153
Dragon: (20) pri=0
2005-Feb-21 22:38:56
49.179.149.193
Dragon: (20) pri=0
2005-Feb-21 22:38:56
128.220.177.49
Dragon: (21) pri=0
2005-Feb-21 22:38:56
99.14.69.153
Dragon: (21) pri=0
2005-Feb-21 22:38:56
99.14.69.153
Dragon: (21) pri=0
2005-Feb-21 22:38
99.14.69.153

Figure 5: The VisAlert system

We employ an *alert beam* indicator to encode multiple instances of the same alert type during the same history period. The alert beam's width represents the number of instances and thus shows the persistence of the problem. The color of the beam encodes the severity of the problem, such as the priority of a *Snort* alert.

Additional features that are not obvious in a static diagram include:

- Interactive information about alerts and nodes (note popup windows in Figure 5).
- Pan and zoom operations over the topology map. In essence the inner ring behaves as a circular window through which the user see portion of the entire topology map. The user can then move the underlying map and zoom in/out to a provide an interactive level of detail (LOD).
- Hierarchical representation of the alert types (LOD over types).
- The history periods, *i.e.*, the rings, can each be be dynamically configured to represent different time periods (LOD over the time domain).
- Animated interaction when opening or closing a group of alert types. This is an important feature that allows the user to maintain awareness of where each alert type is on the screen as their distribution around the ring changes.
- Recording and playback of a session for forensic analysis.

More information and details about *VisAlert* can be found in [13].

4 Visualization Paradigm for Situational Awareness

The *VisAlert* work, discussed in the previous section, targeted a very specific area within the domain of SA; namely, that of network alerts and network topology. We described *VisAlert* with respect to a framework that is based on the *time*, *node*, and *type* attributes. We now revisit this framework and generalize it to pertain to a much larger class of SA applications.

4.1 Events, Resources and the w^3 Premise

Alert correlation systems, such as *VisAlert*, aim to establish the validity of, or to generate a confidence measure for, the participating alerts. The main problem in correlating alerts from disparate logs is the *seeming* lack of mutual grounds on which to base any kind of comparison between the alerts. *VisAlert* takes advantage of the fact that every alert has a *type*, a *time*, and a *node* attribute, which forms a consistent base for comparison. However, this base is not specific to the network alerts nor is it coincidental.

Consider a generic event in *space-time*. By its very nature it must possess what we term the w^3 premise, namely, the *What*, *When*, and *Where* attributes. *When* refers to the point in time when the event happened. *Where* refers to its location. Finally, *What* refers to some global indication of the type of the event. Network alerts are thus a generic event with the mapping of (*What*, *When*, *Where*) to (*type*, *time*, *node*). Furthermore, the network *node* can be thought of representing a local *resource* which is of interest for our situational awareness.

4.2 Generalization Part I

In *VisAlert* we used two fixed functions (hardcoded)

$$f : (type, time) \to (angle, ring)$$
$$g : (node) \to (x, y)$$

For each alert instance in the database, we first extracted only the *type*, *time* and *node* attributes. We then used the f function to find out its location $p_0 = f(type, time)$ on the rings. For alert instances that fall on the innermost ring ($ring = 0$), we used the function g to find its projection onto the topology map $p_1 = g(node)$ and drew a line $\overline{p_0 p_1}$. If the p_0 was outside the last ring we ignored the alert, otherwise we incremented a counter $num[p_0]$.

Using the w^3 premise, we can now generalize the display design of *VisAlert*. We formalize this generalization by defining an event instance as an *n-tuple*,

$$\vec{e} = (What, When, Where, \ldots) \quad (1)$$

That is, an event is an ordered list of attributes where the first three attributes are *What*, *When*, and *Where*. The rest of the attributes are not important within our w^3 framework. We will return to this assumption and generalize it in Section 4.4

Next, we define three unknown functions:

$$\begin{aligned} \theta &: what \to angle \\ \rho &: when \to ring \\ \chi &: where \to (x, y) \end{aligned} \quad (2)$$

We can now combine equations 1 and 2 and define the projection,

$$\Gamma(what, when, where) = \\ = \begin{cases} \overline{p_0 p_1} & \rho(when) = 0 \\ \mathrm{num}[p_0]{+}{+} & 0 < \rho(when) \end{cases} \quad (3)$$

where,

$$p_0 = (\theta(what), \rho(when))$$
$$p_1 = \chi(where)$$

Effectively, we decouple the projection and rendering phases from the mapping phase using the three external functions θ, ρ, and χ. The visualization process is thus made of three separate phases.

1. *Retrieval* of the $(type, time, resource)$ attributes from the database.

2. *Mapping* these values using three external functions.

3. *Projection* onto the display using a generic Γ function.

From an implementation point of view, the three mapping functions can be supplied to the program via a plugin mechanism, while the resources map (the topology map in the case of *VisAlert*) is nothing more than an image file.

Our new visualization paradigm, therefore, provides a general framework for correlating disparate events with respect to a collection of resources.

4.3 Event Correlation

The reader may have noticed that the presentation of the *VisAlert* visualization paradigm in Section 3 was in effect a special case of the event projections we formalized in the previous section. In the case of network alerts, the *Where* attribute (resources) consists of computers and switches, which in turn are represented by a network topology map. However, using the generalized mapping $\theta(type), \rho(time)$, and $\chi(resource)$ and the generic projection Γ, we can apply our visualization paradigm to other areas where situational awareness relies on correlations between events and resources. For each such case we need to

1. Identify the *resource* and *type* domains.

2. Create an appropriate resource map (image).

3. Define the $\theta(type), \rho(time)$ and $\chi(resource)$ plugins.

Our visualization engine can now be applied to a wide variety of SA areas without modifications to the engine itself.

In the following examples, we demonstrate the generality of our visualization paradigm by applying it to various SA areas. For clarity, we refer to all these various visualization solutions under the same name, *VisAware* (thus tying it back to the original *VisAlert* work).

4.3.1 Example: Network Alert Incident Reporting

The first example relates to the original application of *VisAlert*, namely network intrusion alerts. In very large organizations there may be numerous operators that monitor the network at the alert level. While *VisAlert* can fulfill the needs of these operators, they are only responsible to report any suspicious activity. It is the responsibility of higher level analysts to collect these reports and determine if any action is warranted. In some cases there may even be a third level of analysts who oversee the second tier.

It is clear that the second and third tier analysts cannot monitor the entire local network infrastructure. Furthermore, these analysts need to be able to correlate not only network alerts (as per the report they get from the lower level operators) but they also need to correlate the actual reports to each other. By correlating the reports, these analysts can detect broad range attacks such as simultaneous attack on distanced installations, or a set of attacks that repeats itself at different times or at different places.

To adapt *VisAware* to this scenario, we need to define the *Where* (resources) and *What* (types) domains. The *Where* domain in this case includes all the various network installations that a higher level analyst monitors, while the *What* domain includes the various types of reports an operator may submit. Figure 7 shows a possible design of a network alert incident reporting application.

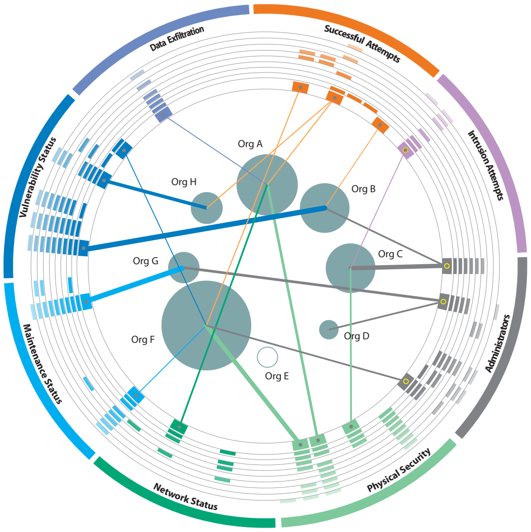


Figure 7: Network alert incident reporting.

4.3.2 Example: Biowatch

In order to protect US citizens from the threat of chemical and biological attacks, the Department of Homeland Security has proposed a program called BioWatch to detect and report the presence of harmful agents. One of the most significant obstacles BioWatch faces is how information about possible attacks is quickly and effectively communicated to control centers.

This information includes:

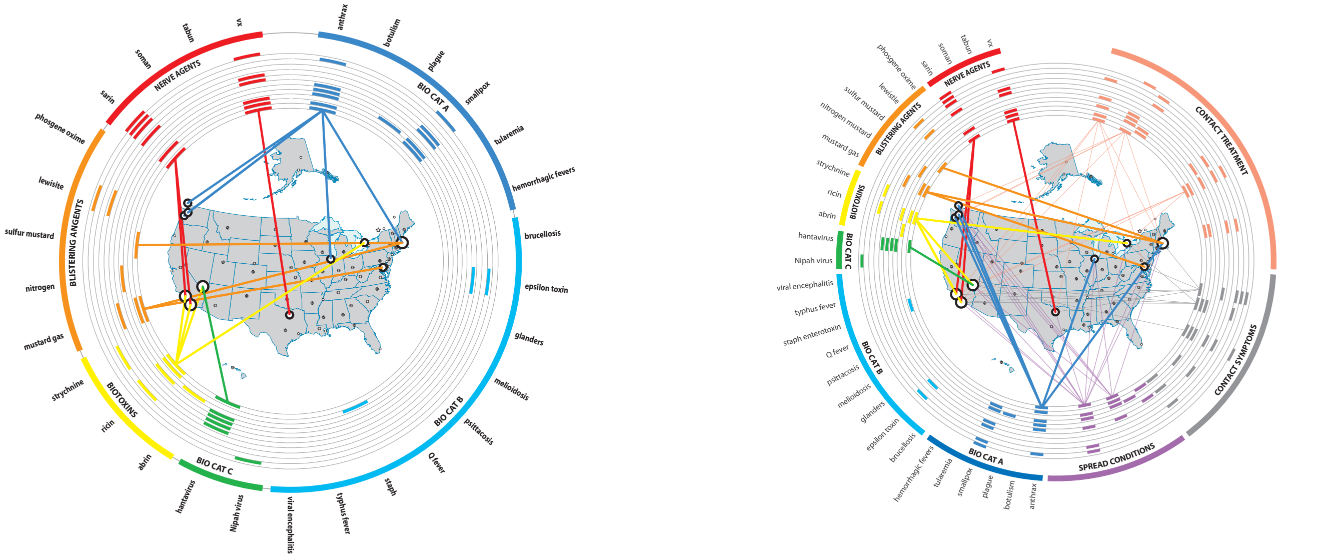
- Detected agent or agents,

- Probability of their existence,

- Where they are detected,

- How the probable presence of such agents changes over time.

Understanding all of the dimensions of this information is critical to developing response plans in order to protect people in danger, and to create defense strategies against future possible attacks.



(a) Incidents

(b) Incidents, Symptoms, and Conditions

Figure 6: VisAware BioWatch showing (a) incidents of agent detection and (b) correlated contact symptoms, treatments, and spread conditions

BioWatch Data The most basic elements of chemical and biological attacks that analysts need to understand are:

- What is the probability that it is an actual attack.
- What types of agents and specific chemical or biological agents are detected.
- Where is the incident.

Beyond this crucial information, other data can help analysts understand the nature of the threat, including:

- How the agents spread (e.g., airborne, water-borne, personal contact),
- Human symptoms of contact,
- Treatment strategies.

The variable relationships and details that different analysts may or may not want to see require that the methods for communicating this information be simultaneously flexible and precise.

VisAware for BioWatch The VisAware structure classifies agents in colored sections around a ring. Figure6(a) shows the different categories of biological agents (as classified by the Centers for Disease Control) and the different types of chemical agents (i.e. blistering and nerve agents). The concentric rings in the circle represent sequential time samples. This modified tree-ring shows how the presence of agents has evolved over time. The inside of the ring structure shows where sensors across the country are set up. With the map in the middle, it is easy to correlate the presence of agents to the sensor that detected it. The correlating line has a variable width that shows the probability of the agent under analysis; the thicker the line the greater the probability of an actual attack. Additional information can be shown on the ring to support more complex analysis. For example, if an analyst wants to understand how symptoms and treatment correlate to the possible presence of an agent, two sections can be added to the ring to include this information as shown in Figure 6(b).

4.4 Generalization Part II

In Section 4.2 we propose viewing an event as an *n-tuple*, or a vector of attributes. That view was derived from the way information is stored in relational databases, namely tables where each row corresponds to an event, and each column represents a specific attribute (whether the events are actually stored in a single table or are recovered via a join operation of several tables is of no importance in our case). The fact that different events from different tables may have different attributes was ignored, based on the w^3 premise, and was hidden away by the use of a retrieval package that selected from the database only the first three columns of each table, corresponding to the *What*, *When*, and *Where* attributes.

We can generalize the *n-tuple* view of an event by introducing the notion of a *property list*. A property list is an unordered collection of $(key, value)$ pairs. Using the w^3 premise, we can represent an event as a special property list that includes at least the three *What*, *When*, and *Where* keys:

$$\vec{e} = \{(key, value) \mid \exists i, j, k \quad key_i = What,\ key_j = When,\ key_k = Where \tag{4}$$

Let us also define a generic projection:

$$\Pi_{key}(\vec{e}) = \vec{e}(key)$$

That is, for a given attribute key , $\Pi_{key}(\vec{e})$ returns the value of $\vec{e}$ associated with that attribute.

The three mapping functions in equation 2 can now be reformulated as follows:

$$\begin{aligned} \theta'(\vec{e}) &= \theta \circ \Pi_{what}(\vec{e}) \\ \rho'(\vec{e}) &= \rho \circ \Pi_{when}(\vec{e}) \\ \chi'(\vec{e}) &= \chi \circ \Pi_{where}(\vec{e}) \end{aligned} \tag{5}$$

It is clear from equation 5 that the *What*, *When*, and *Where* are not intrinsic to our formulation. They are only leftover artifacts of our original *VisAlert* design and the w^3 premise from Section 4.1. We can reformulate the three mapping again, this time taking out the last remnants of the w^3 premise.

$$\begin{aligned} \theta^{\alpha}(\vec{e}) &= \theta \circ \Pi_{\alpha}(\vec{e}) \\ \rho^{\beta}(\vec{e}) &= \rho \circ \Pi_{\beta}(\vec{e}) \\ \chi^{\gamma}(\vec{e}) &= \chi \circ \Pi_{\gamma}(\vec{e}) \end{aligned} \tag{6}$$

That is, given *any* three attributes α, β, γ, and *any* three mapping function θ, ρ, χ, we can define our generic projection,

$$\Gamma^{\alpha,\beta,\gamma}(\vec{e}) = \begin{cases} \overline{p_0 p_1} & \overline{\rho} = 0 \\ \text{num}[p_0]{+}{+} & 0 < \overline{\rho} \end{cases} \tag{7}$$

where,

$$\begin{aligned} \overline{\theta} &= \theta^{\alpha}(\vec{e}) & p_0 &= (\overline{\theta}, \overline{\rho}) \\ \overline{\rho} &= \rho^{\beta}(\vec{e}) & p_1 &= \overline{\chi} \\ \overline{\chi} &= \chi^{\gamma}(\vec{e}) \end{aligned}$$

Now that the mapping functions do not depend on any specific attribute, we can make the final step in our generalization process, and extend them to operate on any collection of attributes. More formally, we represent a collection of attributes as:

$$\vec{\alpha} = (\alpha_0, \alpha_1, \ldots, \alpha_n)$$

and define the projection:

$$\begin{aligned} \Pi_{\vec{\alpha}}(\vec{e}) &= (\Pi_{\alpha_0}, \Pi_{\alpha_1}, \ldots, \Pi_{\alpha_n})(\vec{e}) \\ &= (\Pi_{\alpha_0}(\vec{e}), \Pi_{\alpha_1}(\vec{e}), \ldots, \Pi_{\alpha_n}(\vec{e})) \end{aligned}$$

and replace (α, β, γ) in equations 6 and 7 with $(\vec{\alpha}, \vec{\beta}, \vec{\gamma})$. That is, the three projections are each defined with respect to one or more attributes, and their value depends on some kind of combination of the values of these attributes for the given event, $\vec{e}$.

Our final visualization paradigm, depicted in Figure 8,provides a general framework for correlating disparate events with respect to any combination of attributes. We now present a few examples of such a general visual correlation of events for situation awareness.

4.5 Example: Emergency Centers

Emergency forces such as the police, fire department, 911 emergency centers, and recently the Department of Homeland Security are prime examples of areas where situational awareness is a vital component of their everyday work. For the people in the field, SA basically means being aware of the environment they are embedded in. For these cases, our visualization paradigm may not be the best solution. However, *VisAware* does fit well for the people at emergency centers who may need to be aware of large number of simultaneous events of different emergency levels and at various locations. Furthermore, each event may include not only several simultaneous emergencies, such as fire and wounded people, but may require participation by many different units and forces.

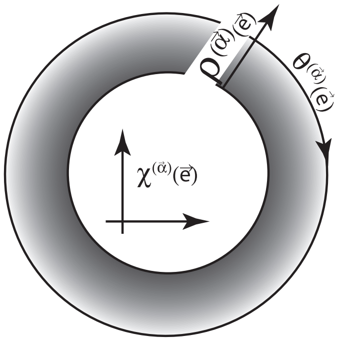

$\rho^{(\vec{\alpha})}(\vec{e})$
$\theta^{(\vec{\alpha})}(\vec{e})$
$\chi^{(\vec{\alpha})}(\vec{e})$

Figure 8: Generalized visual correlation of events for situational awareness

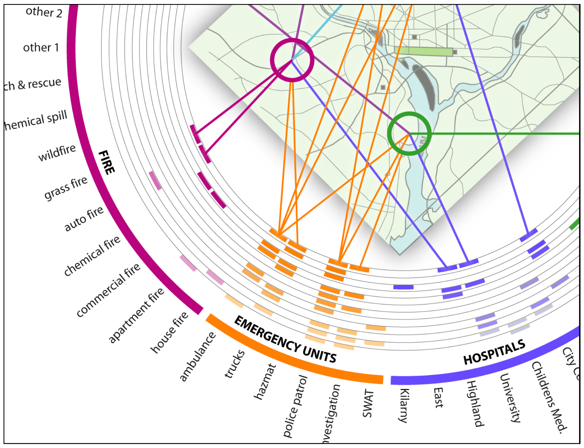


Figure 9: Emergency awareness: Note that the ring includes both event types *and* resources such as hospital and emergency units (police patrols, ambulances)

With respect to our generalized framework, the $\rho(\vec{e})$ function is similar to the one in Examples 4.3.1 and 4.3.2. The domain of the $\theta(\vec{e})$ on the other hand can now include resources, such police and other emergency units on or moving to the scene, as well as event types, such as fire and traffic occident's. Hospitals can also be listed on the ring to show where victims are being sent to. This can be especially important in major incidents were the victims are sent to various hospitals around the city. Finally, the the range of the χ function may be a map of a city, as depicted in Figure 9.

5 Conclusions and Future Work

In this paper we describe a flexible visualization framework for increase situational awareness in various SA areas. We provide example of application of our framework to several

SA areas. We have not formally tested our framework, although *VisAlert*, the visualization for network intrusion detection, was alpha tested at Air Force Research Lab (AFRL) in Rome, New York, with very favorable initial reaction. We are currently planning to install *VisAlert* in several beta sites to systematically measure SA. We are also working on *VisAware*, the second generation of *VisAlert*, that is based on the generalized framework presented in this paper.

There are several techniques for measuring SA [4]. These techniques can be loosely defined as objective measures, such as the Situation Awareness Global Assessment Technique (SAGAT), subjective measures such as the Situational Awareness Rating Technique (SART), and performance-based measures. The SAGAT asks probing questions about information displayed on the visualization during temporary halts to the user's activity and blanking the screen. The SART provides a subjective measure of SA by the users. The users rate their perception on a variety of scales such as demand and understanding. Performance-based measures provide an outcome-based metric to assess the SA of the user.

Each of these techniques has its advantages and disadvantages. The SAGAT provides an objective measure of the user's ability to obtain information from the display; however, the main disadvantage is the reliance upon the user's memory and therefore the results may include a memory bias. The main advantage of SART is that it is very easy to implement; however, it may be influenced by the users self-reporting bias. In addition to the SART and SAGAT scores, performance data will also be collected to provide an action-based assessment of the SA differences between user groups. Although scores on the SART and SAGAT are often not correlated [9], in this proposed investigation, SART and SAGAT scores will be compared between groups of users who use traditional tools and groups that use the VisAlert system.

We plan to use all three methods for assessing SA in a simulated network operational environment to probe users' ability to accurately understand the information that is presented to them. We plan to simulate several attack scenarios and then measure how accurately and quickly they are able to extract this information from the display by stopping the simulation at predetermined intervals and asking questions about the state of the network, how many alerts are associated with particular nodes, and what alerts are the most prevalent. Users will then take an appropriate action that will serve as the performance measure. In addition, at the end of the simulation we will use the SART method to obtain indications of their perceived workload and performance. These numbers will be compared to the control group that receives the same scenarios and questions but who use the traditional tools.

Acknowledgments

We would like to thank ARFL's Rome labs for hosting a field test of the VisAlert system. We owe special thanks to Kirsten Whitley whose support of the VisAlert project was instrumental. Many thanks go to Geoff Draper, Fernando A. Iribarren and Xavier Tricoch for their valuable comments and suggestions. This work was supported in part by a grant from the IC-ARDA.

References

[1] Hervé Debar and Andreas Wespi. Aggregation and correlation of intrusion-detection alerts. In *Recent Advances in Intrusion Detection*, pages 85–103, 2001.

[2] Secure Decisions. http://www.securedecisions.com.

[3] M. R. Endsley. Design and evaluation for situation awareness enhancement. In *Proceedings of the Human Factors Society 32nd Annual Meeting*, pages 97–101, Santa Monica, CA, 1988. Human Factor Society.

[4] M. R. Endsley, R. Sollenberger, and E. Stein. Situation awareness: A comparison of measures. In *Proceedings of the Human Performance, Situation Awareness and Automation: User-Centered Design for the New Millennium*, Savannah, GA, 2000. SA Technologies, Inc.

[5] Mica R. Endsley, Betty Bolté, and Debra G. Jones. *Designing For Situational Awareness, An Approach to User-Centered Design*. Taylor & Francis, 2003.

[6] Deborah Estrin, Mark Handley, John Heidermann, Steven McCanne, Ya Xu, and Haobo Yu. Network visualization with nam, the vint network animator. *IEEE Computer*, 33(11):63–68, November 2000.

[7] Michels P. Gravenstein and D. Westenskow. Dr: An integrated graphic data display improves detection and identification of critical events during anesthesia. *J. Clin. Monit*, 13:249–259, 1997.

[8] D. G. Jones and M. R. Endsley. Sources of situation awareness errors in aviation. *Aviation, Space and Environmental Medicine*, 67(6):507–512, 1996.

[9] D. G. Jones and M. R. Endsley. Can real-time probes provide a valid measure of situation awareness? In *Proceedings of the Human Performance, Situation Awareness and Automation: User-Centered Design for the New Millennium*, Savannah, GA, 2000. SA Technologies, Inc.

[10] Clarence A. Robinson Jr. A powerful vision. Signal Magazine, August 2001. http://www.afcea.org/signal/articles.

[11] Stephen Eick Kenneth Cox and Taosong He. 3d geographic network displays. *ACM Sigmod Record*, 25(4), 50 1996. December.

[12] Kiran Lakkaraju and abd Adam J. Lee William Yurcik. Nvisionip: netflow visualizations of system state for security situational awareness. In *Proceedings of CCS Workshop on Visualization and Data Mining for Computer Security, ACM Conference on Computer and Communications Security*, October 29 2004.

[13] Yarden Livnat, Jim Agutter, Shaun Moon, Robert F. Erbacher, and Stefano Foresti. A visualization paradigm for network intrusion detection. In *6th IEEE Systems, Man and Cybernetics Information Assurance Workshop*, pages 92–99, West Point, NY, June 2005. IEEE.

[14] Jonathan McPherson, Kwan-Liu Ma, Paul Krystosekand Tony Bartoletti, and Marvin Christensen. Portvis: A tool for port-based detection of security events. In *CCS Workshop on Visualization and Data Mining for Computer Security*, October 2004.

[15] Quantum3D. http://www.quantum3d.com/stories/isd.htm.

[16] K.L. Ma S.T. Teoh and S. F. Wu. Visual exploration process for the analysis of internet routing data. In *IEEE Conference on Visualization 2003*, pages 523–530, 2003.

[17] Alfonso Valdes and Keith Skinner. Probabilistic alert correlation. In *Recent Advances in Intrusion Detection*, pages 54–68, 2001.

[18] Alex Wood. Intrusion detection: Visualizing attacks in ids data. Giac gcia practical, SANS Institute, February 2003.

[19] Xiaoxin Yin, William Yurcik, Michael Treaster, Yifan Li, and Kiran Lakkaraju. Visflowconnect: netflow visualizations of link relationships for security situational awareness. In *Proceedings of CCS Workshop on Visualization and Data Mining for Computer Security,ACM Conference on Computer and Communications Security*, October 29 2004.