Secure & Resilient Cyber Architectures Conference
MITRE, McLean, VA, October 29, 2010

A Semantic Architecture for Enhanced Cyber Situational Awareness

John Strassner[1], Joseph Betser[2], Roberta Ewart[3], Frank Belz[2]
[1] DENSoft Corporation, Chicago, Illinois; john@densoft.org;
[2] The Aerospace Corporation, El Segundo, California; {betser, frank.c.belz}@aero.org;
[3] USAF Space and Missile Systems Center, El Segundo, California; roberta.ewart@losangeles.af.mil;

***Abstract*—The cyber analyst must try to sift through a huge amount of data that may or may not be related in order to identify threats. This is a complex process that is made more difficult by having to correlate and combine heterogeneous data that are created using different languages with varying amounts of semantics. However, data alone is insufficient to identify and assess threats; behavior must also be observed and inferred, so that appropriate actions can be taken. This paper proposes a new Cyber Situational Awareness architecture that draws from elements of established situation awareness, decision-making, and data and knowledge fusion models. While based on the Observe-Orient-Decide-Act loop, it adds a number of concepts required to better support cyber situational awareness.**

***Index Terms*—C4ISR, data fusion, JDL model, information fusion, OODA, situational awareness**

I. Introduction

In today's complex world, the cyber-analyst is inundated with large amounts of diverse information that may contain threat indicators. Often, a threat can only be determined by correlating indicators from multiple sources that provide different types of data (e.g., voice and video). In addition, such data may be in different formats and languages, be accessible from different protocols and systems, and be from different locations; these and other factors make it difficult to combine different threat indicators into a single comprehensive situational view. The scope of the threat can be generic in nature or tied to one or more missions. Threats can attack a mission directly and/or indirectly, such as by compromising the functionality of one or more support entities of the mission.

An even more challenging problem is to analyze the threat indicators to infer what actions the attackers could take, and what their ultimate goals are. As the nature of cyber threats continues to evolve, it is imperative for the cyber analyst to be able to quickly identify threat indicators, anticipate their associated targets, and assess, by fusing data from multiple sources and locations, the damage successfully executed threats would cause.

A widely accepted definition of Situational Awareness is ~~t~~the perception of elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future" [1]. This paper modifies this definition to facilitate focusing on the cognitive aspects of situational awareness, as follows: ~~t~~the perception of data and behavior that pertain to the relevant circumstances and/or conditions of a system or process (~~t~~the situation"), the comprehension of the meaning and significance of these data and behaviors, and how processes, actions, and new situations inferred from these data and processes are likely to evolve in the near future to enable decision-making superiority". This definition makes the following explicit: (1) a situation can be determined not just by data, but by behavior, (2) both data and behavior are important for recognizing and understanding a situation, (3) the evolution of future situations can be inferred through data and/or behavior, and (4) the reason for situational awareness is to enable making better decisions.

Situation awareness requires the ability to identify and analyze domain-specific and cross-domain data and behavior, and relate each to the situation(s) of interest. In today's cyber world, the context of a situation can vary both the meaning of observed data and behavior as well as the underlying mission to be accomplished, which in turn can influence the actions that should be taken.

While human models of situational awareness are well developed, machine-understandable models are still under active development. This paper proposes a new architecture that is focused on the cognitive aspects of situational awareness for C4ISR tasks. It is motivated by the merging of three established models: Endsley's Situation Awareness model [1], the JDL Data Fusion model [2], and Boyd's OODA loop [3]. Three improvements to current situational awareness models are proposed. First, a detailed model of context is used that supports self-awareness, enabling the system to control information overload as well as treat threat indicators appropriately on a per-situation basis. Second, different situations dictate different ways of gathering, analyzing, and reacting to data; hence, a robust and extensible policy model is used to govern situated actions that can process as well as react to data. Finally, extensions to Boyd's well known Observe-Orient-Decide-Act (OODA) decision cycle are integrated with the JDL Data Fusion model and Endsley's model of situation awareness to better represent the characteristics and behavior required by situation aware machine based applications for cyber threat identification

and management. This enables decision-making to stay focused, enabling it to adjust to changes in the environment.

These improvements are realized as extensions to the FOCALE autonomic architecture [4], which was developed to support context-aware, ubiquitous computing applications. FOCALE uses an enhanced form of the OODA loop, but has been up to now focused on autonomic network applications that require self-awareness [5]. We believe that this provides a strong foundation for cyber situational awareness (CSA), and focus on some important modifications to this architecture for CSA applications.

The organization of this paper is as follows. Section II describes the established models that we are drawing from (Endsley's Situation Awareness model, the JDL data fusion model, and Boyd's OODA loop). Section III briefly reviews the FOCALE autonomic architecture, emphasizing its novel context-aware policy governance model and its underlying cognition model. Section IV explains our new architecture, and Section V describes future work.

II. Elements of Situation Awareness

This section briefly describes the important and established work that we are drawing from in building our new architecture.

A. Situation Awareness

Endsley's Situation Awareness model is shown in Figure 1, and consists of three levels. The first, Perception, produces an awareness of situational elements (e.g., objects, events, people, systems, and environmental factors) and their current states (e.g., modes and locations). The second, Comprehension, examines the level 1 elements to better understand how they fit together; this helps characterize the situation as a whole, and how this affects the goals of the mission. The third, Projection, is focused on predicting the most likely evolution of the situation.

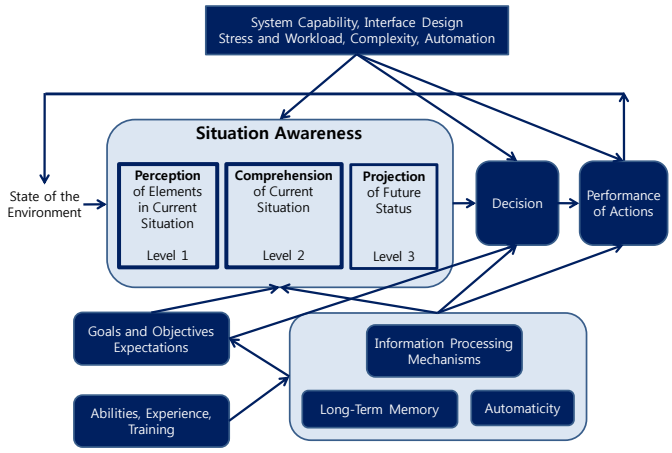


Figure 1. Endsley's Model of Situation Awareness

In Endsley's model, situation awareness corresponds to the operator's internal model of the state of the environment, which is separate from decision making and performance. This is due to many reasons; arguably, the most important is that even if there is a perfect understanding of the situation, incorrect decisions can still be made, due to organizational or technical constraints, lack of experience, or human factors such as impulsiveness or risk aversion.

This model also reflects the differences between working, short-, and long-term memory from cognitive psychology, which is used to guide the FOCALE architecture; this is described more in Section III.

B. Fusion Models

Data fusion is necessitated when heterogeneous systems are required to interoperate in an open world, where the syntax and semantics of data provided by a sensor or a human is domain-specific and does not conform to any one specific vocabulary. This requires the translation of each vocabulary used into a common set of concepts and terms, so that the data can be integrated.

Many different fusion models have been proposed. One of the most cited and still prevalent is the JDL Data Fusion model [2] [3], which is a reference model that describes the overall process of combining data from varied sources to result in better understanding of the situation being observed. It can be divided into five levels: Level 0, Feature Assessment (raw data capture and processing), Level 1, Entity Assessment (entity and event identification and tracking), Level 2, Situation Assessment (relational analysis of objects and events to form situations), Level 3 Impact Assessment (threat intent estimation and consequence prediction), and Level 4, Performance Assessment (resource management). At each level, algorithms can be applied to make inferences about the meaning of the data in context. It is this inference capability that gives the data fusion system its power. The JDL model is shown in Figure 2.

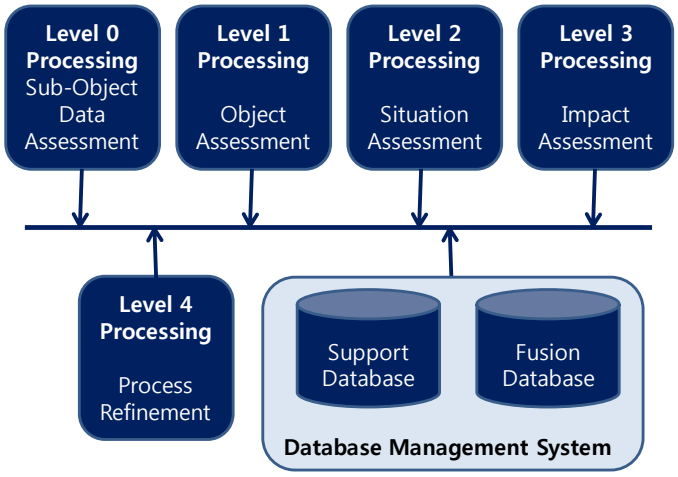


Figure 2. JDL Data Fusion Model

Situational awareness corresponds to Level 2 of the JDL Data Fusion Model, and is an essential step towards understanding and safe-guarding the cyber-infrastructure. However, situational awareness in general and the JDL Data Fusion model in particular are relatively weak at identifying behavioral patterns that can be associated with different types of cyber attacks. This prevents systems based on such models from

learning from experience as well as dynamically adapting to attack patterns.

In addition, the JDL Data Fusion model is inherently data focused. While some threats can be inferred by data analysis, others cannot, and instead rely on recognizing behavior to indicate a particular situation. In addition, the wary attacker will usually try and obfuscate the data or mislead detection systems; this is easier to do with false data than false behavior.

C. Decision-Making

Col. John Boyd's control loop [3] consists of four phases: Observe, Orient, Decide and Act. It is shown in Figure 3.

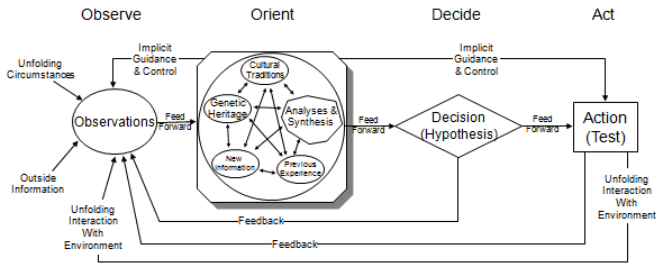


Figure 3. Boyd's OODA Loop

Figure 3 is drawn to emphasize how orientation shapes observation, decision, and action. While the loop appears to be sequential, this is merely for convenience. Observation, orientation, and action occur simultaneously and continuously. In reality, people usually do not employ the explicit, sequential O-to-O-to-D-to-A mechanism. Instead, they simply observe and act. In either case, the orientation step is critical, as it determines how observations, decisions, and actions are performed. As Boyd observed, people act according to how they perceive the world, as opposed to how the world really is.

One of the strongest features of the OODA loop is to initiate or modify actions in the light of observed events. If this can be transformed into a machine-understandable form, then logic can be applied to examine all different concurrent options to arrive at the best plan to achieve the goals of the mission.

The OODA loop does have some deficiencies. For example, the opponent is not explicitly shown, and the model needs some modifications for collaborative decision-making (e.g., shared situation awareness, task re-allocation, negotiation of goals, confirmation and authorization of decisions). Finally, attention and memory, as well as a cognitive representation of the world, need to be added. Note that this is not meant to imply that the OODA loop should not be used; rather, our aim is to generalize it to make it usable beyond its original intended audience.

III. The FOCALE Autonomic Architecture

FOCALE, which stands for Foundation – Observe – Compare – Act – Learn – Reason, was created to automate the complex, manually-intensive configuration tasks of network devices. The the main components of FOCALE are shown in Figure 4. Each building block is connected using a distributed and enhanced Enterprise Service Bus (ESB) [6] that supports simple as well as semantic queries. The difference between it and standard ESBs is that it can be used to orchestrate content, whereas standard ESBs are limited to orchestrating messages. The FOCALE Autonomic Manager uses the ECB to orchestrate behavior. It can support different types of knowledge acquisition and distribution (e.g., push, pull, and scheduled) and performs common processing (e.g., semantic annotation, filtering and storage) before content is delivered to components. This enables components to register interest in knowledge in a more precise fashion, and thus reduce messaging overhead.

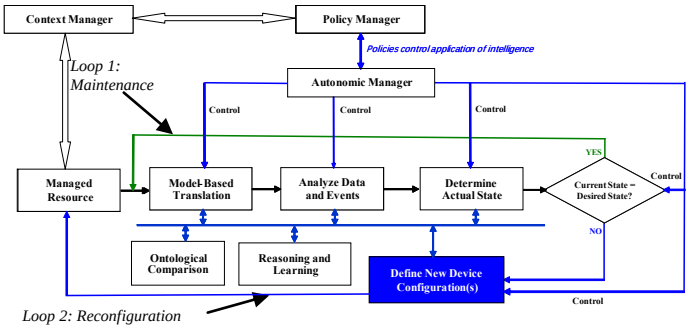


Figure 4. Simplified Version of FOCALE

FOCALE uses the DEN-ng information model [7] and the DENON-ng [8] ontologies to translate disparate sensed data into a common networking lingua franca. DEN-ng is used to represent the static characteristics and behavior of entities; DENON-ng is then used to augment this model with consensual meaning and definitions so that domain- and vendor-specific concepts can be mapped into a common terminology. This enables facts extracted from sensor input data to be reasoned about using ontology-based inferencing.

In FOCALE, sensor data is retrieved and translated from vendor- and device-specific data into a normalized form in XML using model-based mapping and ontology-based reasoning. This is then analyzed to determine the current state of the managed entity. The current state is compared to the desired state from the appropriate Finite State Machines (FSMs). If no problems are detected, the system continues using the maintenance loop; otherwise, the reconfiguration loop is used so that the services and resources provided can adapt to these new needs.

Nodes in a FOCALE FSM represent a configuration state; edges represent state transitions, and connote permission to change the configuration of a managed resource. Static behavior is thus ~~"p~~programmed" into FOCALE by designing a set of FSMs; dynamic behavior is defined by altering one or more FSMs. Context-aware policy rules govern both autonomic control loops [9]. This enables context to select the set of policies that are applicable; policies are used to then define the functionality allowed. The autonomic manager uses the current set of context-aware policies to govern each of the architectural components of the control loop, enabling each of the different control loop components to change how it operates as a function of context. As context changes, policies change, and system functionality is adjusted accordingly.

There are a number of similarities between FOCALE and OODA. For example, the need to orient observations is the inspiration for the FOCALE model-based translation layer, which orients observed data to the current context and translates different data sources into a single neutral form to facilitate their correlation and integration. Like OODA, FOCALE is not a sequential loop. First, it is not a good idea to stop observing while the analysis is continuing. Second, a balance must be maintained between delaying decisions (which means delaying actions) and performing more accurate analysis that eliminates the need to revisit previously made decisions. Third, both the speed of re-orientation as well as being able to apply suitable actions via the implicit guidance and control link to Action are critical for supporting decision-making. This enables simpler control loops to be employed in situations that warrant it.

IV. Semantic Cyber Situational Awareness

This section provides an overview of our evolving CSA architecture.

A. Comparing FOCALE and OODA

The original FOCALE control loops are based on the OODA loops, as shown in Figure 5. The Model-Based Translation function transforms raw sensor data into a form that can be correlated with other sensor data from the current context. It is represented by the Normalize function in Figure 5. It, along with the Analyze and Determine State function, correspond to the Orient process. The Compare function of FOCALE corresponds to the Decide function of OODA, except that OODA is not focused on state, whereas FOCALE is; this is because FOCALE uses state to orchestrate behavior. This is reflected in the Foundation function of FOCALE as well.

The next two sections describe how cognition is used to enhance this architecture. This provides a framework for adding semantic reasoning to FOCALE.

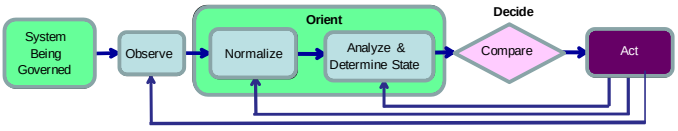


Figure 5. Comparing OODA and FOCALE

B. Cognition

A cognitive system is one that can reason about what actions to take, even if a situation that it encounters has not been anticipated. It can learn from its experience to improve its performance. It can also examine its own capabilities and prioritize the use of its services and resources, and if necessary, explain what it did and accept external commands to perform necessary actions. Endsley's model supports cognition in several important ways. Its perception portion provides the notion of classifying data into pre-defined representations that are understood and relevant to the current situation. Memory is used to increase comprehension of the situation. Finally, actions are judged by how effectively they perform to support the situation. These concepts are supported in cognitive psychology, where Minsky modeled this using three interacting layers, called reactive (or subconscious), deliberative, and reflective [10].

Reactive processes take immediate responses based upon the reception of an appropriate external stimulus. Such processes have no sense for what external events ―mean"; rather, they simply respond with some combination of instinctual and learned reactions.

Deliberative processes receive data from and can send ―commands" to the reactive processes; however, they do not interact directly with the external world. This part of the brain is responsible for our ability to achieve more complex goals by applying short- and long-term memory in order to create and carry out more elaborate plans. This knowledge is accumulated and generalized from personal experience and what we learn from others.

Reflective processes supervise the interaction between the deliberative and reactive processes. These processes enable the brain to reformulate and reframe its interpretation of the situation in a way that may lead to more creative and effective strategies. It considers what predictions turned out wrong, along with what obstacles and constraints were encountered, in order to prevent sub-optimal performance from occurring again. It also includes self-reflection, which analyzes how well the actions that were taken solved the problem at hand.

The latest iteration of FOCALE has a new cognition model that reflects the above differences, and is shown in Figure 6.

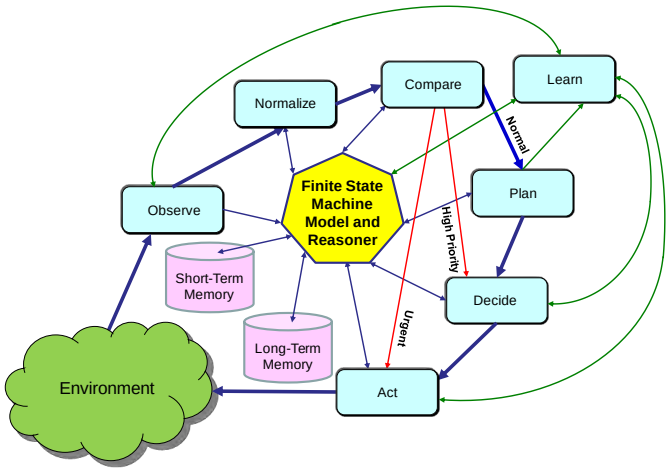


Figure 6. The New FOCALE Cognition Model

All processes use the FSM and reasoner, which enable the system to infer results from incomplete data. The reasoner is used by different machine learning algorithms to realize when an event or set of events has been previously encountered. Such results are stored in short-term memory. When a previously encountered situation is recognized, the system can bypass many of the computationally intensive portions of the control loop, producing the two ―shortcuts" labeled ―High Priority" and ―Urgent". This enables the system to respond in an agile manner if it recognizes a previous situation (or events leading to one).

The deliberative process is embodied in the set of bold arrows,

which takes the Observe-Normalize-Compare-Plan-Decide-Act path. This uses long-term memory to store how goals are met on a context-specific basis. The reflective process examines the different conclusions made by the set of deliberative processes being used, and tries to predict the best set of actions that will maximize the goals being addressed by the system. This process uses semantic analysis to understand why a particular context was entered and why a context change occurred to help predict how to more easily and efficiently change contexts in the future. These results are also stored in long-term memory, so that the system can better understand contextual changes and explain its reasoning to aid debugging. The reflective path uses the learning functions to perform deeper analyses than those provided by the deliberative path. In our implementation, multiple repositories are used to optimize storage and query processing, but that is beyond the scope of this paper.

In FOCALE, a situation reflects an entity's contextual view of a collection of data and processes at a particular instance in time. Shared situational awareness is therefore a consensus view of a number of individual views that each describes the same situation. Note that the JDL level 2 does not distinguish between past, present and future, while the JDL level 3 is solely focused on the future. In FOCALE, we have added several important features to OODA, including: (1) the notion of *planning*, which corresponds to Minsky's three views of reactive, deliberative, and reflective; (2) the inclusion of *state*, so that a situation can be referenced against a broader world model; (3) the inclusion of *learning*, which supports agile decision-making. This gives rise to the following revised model, shown in Figure 7.

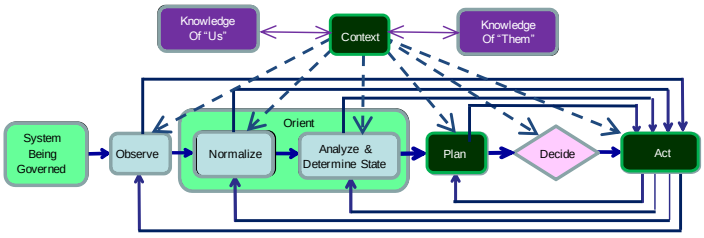


Figure 7. Revised FOCALE Control Loops

Three major enhancements are shown in Figure 7. First, a Plan process is inserted between the Orient and Decide processes. This enables deliberative and reflective processing to occur. Second, feedforward loops are created, so that the Action stage has sufficient information to make informed decisions. This also helps realize the ~~–~~shortcuts" described earlier, which are essential in capturing the original agility that the OODA loop had. Finally, Context is used to choose the most appropriate policy rules for each processing step.

C. Knowledge Representation

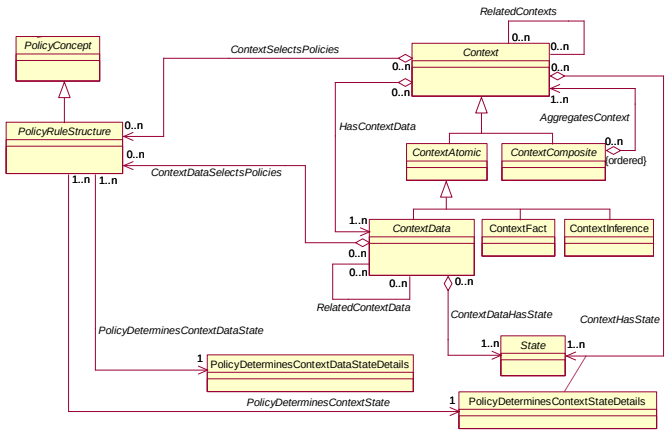
FOCALE defines a concept called the Knowledge Continuum, which provides a mechanism to establish continuity between otherwise disparate viewpoints of knowledge. In particular, this approach rejects the notion of having to define one transformation to harmonize and reconcile different types of knowledge from different sources having different abstractions. In contrast, the Knowledge Continuum asserts that in order to ensure the correct understanding of knowledge at one abstraction, and to be able to relate that abstraction to other views, knowledge itself must be represented in a series of views, where each view has meaning in a specific frame of reference, and where each successive view is generated from a transformation being applied to the preceding view. These views represent the needs and grammars of different constituencies working on a common solution (e.g., business analysts and network technicians), which results in a pure transformation pipeline, which can be implemented in a manner similar to that of [11].

Within each level of the Knowledge Continuum, factual and inferred knowledge can exist that is procedural or declarative. In other words, knowledge is assigned to a particular level in the Knowledge Continuum based on whether it is business or technical in nature, and whether it is device- and technology-specific or not. The Knowledge Continuum is used to guide the modeling of knowledge, using information models and ontologies, to ensure that all key concepts from all constituencies in a managed system are represented.

Of particular importance to the resilience of the system and its process is how data and behavior are used to identify and understand entities, to enable threat indicators to be recognized as threats, and data and behavior used to predict threat outcomes. In particular, changes in data and context may require changes in one or more processes to successfully accomplish the mission; such changes can affect the type of data collected and how such data are processed in addition to other actions. This context-aware feedback is one of the strengths of our architecture.

D. Context Awareness

As context changes, policies change, and system functionality



is adjusted accordingly. This is shown in Figure 8.

Figure 8. DEN-ng Context-Aware Policy Rules

In DEN-ng, context is defined as an aggregation of different aspects that collectively define an overall context. Each aspect of context, such as time and location, is modeled by a ContextData instance, whereas the Context instance represents the final assembled context with all of its different aspects. Both ContextData as well as Context can affect the set of policy rules

that are currently used to govern behavior; this enables granular decisions that are dependent on one or more aspects of Context to change policy rules as well as a higher level change in the entire context to change policy rules. These two changes can also be linked to the changing of state in one or more FOCALE FSMs; this is provided in a two step process. The first step defines the set of policy rules that are used to determine how context affects a State, and the second defines the set of actions for either maintaining that State or transitioning to a new State. These are defined by the PolicyDeterminesContextStateDetails and the PolicyDeterminesContextDataStateDetails association classes, and by the two associations ContextDataHasState and ContextHasState. Additional details of context-aware policy management are given in [9].

E. Semantic Reasoning for Situation Awareness

Our approach is predicated on defining and incorporating semantics into the framework described in the above sections. Semantics are crucial for all phases of CSA. Due to the potential large volume of data that needs to be processed, we target the use of Semantic Web technologies [12] to help manage the workload. In particular, such technologies are excellent at finding hidden data and patterns of data; these are both important indicators of behavior.

Our architecture uses triple stores, which are databases that are specially designed for the storage and retrieval of data defined using Resource Description Framework, OWL, and similar technologies. We use an ontology for two important purposes: (1) it provides a shared vocabulary for describing concepts and entities of interest, and (2) it permits reasoning over these concepts. An advantage of using an ontology to formally represent concepts of interest is that facts that are not explicitly stated can be derived using an inference engine. This is very important, as it is impossible to explicitly enumerate all possible concepts, let alone their interactions and uses. Furthermore, by defining a taxonomy of concepts as an ontology, we can also specify relationships between different objects to help codify knowledge about each object. This supports specific as well as generic situations through the structure of the taxonomy – more generic characteristics and behavior are defined at the higher levels of the taxonomy, while more specific ones are defined in the lower levels.

Ontologies can be used to reason on concepts as well as instances. Our architecture uses several software architecture patterns to develop inference engines for different tasks, such as support multi-sensor fusion and intrusion detection. This enables us to create an extensible system that can address the needs of specific situations while ensuring that knowledge fusion can be accomplished. Architecturally, we use the pipes-and-filters pattern [14] to enable the consecutive execution of successive components that are then combined to provide an overall end result.

While OWL can be used to accomplish many operations directly, there are other cases for which OWL is not sufficiently expressive to capture all of the desired concepts. For example, it is not possible to construct a complex property defined as the composition of other properties. In such cases, we can augment our ontology by using Semantic Web Rule Language (SWRL) [14] rules.

Semantics aid CSA by supporting hypothesis-driven reasoning. A hypothesis can be used to guide the gathering and processing of information. Hypotheses can be used to process incomplete data, to infer missing data (using backward inference), to check the validity of data, and to reduce false positives. In particular, we use hypotheses to help determine if a threat exists based on the set of observed data and inferences for a given context.

V. Conclusions and Future Work

Situation awareness involve the identification and analysis of data and behaviors of entities of interest, so that exiting knowledge may be leveraged to understand sensed information and observed behaviors relevant to a situation. We have a defined a CSA architecture that supports the integration of data from multiple sources into a common whole. This knowledge-level fusion is designed to be automated through a distributed version of the FOCALE architecture. Future work will include formalizing a CSA ontology and testing our architecture for scalability and reusability.

References

[1] M.R. Endsley, "Toward a theory of situation awareness in dynamic systems", Human Factors 37(1), pages 32-64
[2] A. Steinberg, F. White, C. Bowman, "Revisions to the JDL data fusion model", Proc. SPIE, Vol. 3719, pages 430-441, 1999.
[3] J.R. Boyd, "The Essence of Winning and Losing", 28 June, 1995
[4] J. Strassner, N. Agoulmine, E. Lehtihet, "FOCALE – A Novel Autonomic Networking Architecture", ITSSA Journal, Vol. 3, No. 1, May 2007, pages 64-79
[5] A. Tauber, "The Biological Notion of Self and Non-self", The Stanford Encyclopedia of Philosophy Self-aware
[6] B. Christudas, "Service-Oriented Java Business Integration: Enterprise Service Bus Integration Solutions for Java Developers", Packt Publishing, August 2008
[7] J. Strassner, "Introduction to DEN-ng", Tutorial for FP7 PanLab II Project, January 21, 2009.
[8] M. Serrano, J. Serrat, J. Strassner, M. Ó Foghlú, "Management and Context Integration Based on Ontologies, Behind the Interoperability in Autonomic Communications", extended journal publication of SIWN Complex Open Distributed Systems, Chengdu, China, Vol 1, No. 4, July 2007
[9] J. Strassner, J. de Souza, S. van der Meer, S. Davy, K. Barrett, D. Raymer, S. Samudrala, "The Design of a New Policy Model to Support Ontology-Driven Reasoning for Autonomic Networking", Journal of Network and Systems Management, Volume 17, Issue 1, March 2009
[10] M. Minsky, "The Society of Mind", Simon and Schuster, New York, 1988
[11] A. Reggiori, A. Marchesini, "SPARQL To Objects (S2O) – A SPARQL extension for expressing mappings between RDF graphs and JSON objects", Version 0.41, November 8, 2007
[12] http://www.w3.org/standards/semanticweb/
[13] http://www.ontotext.com/owlim/
[14] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, M. Stal, "Pattern-Oriented Software Architecture—A System of Patterns", Addison-Wesley,1998
[15] http://www.w3.org/Submission/SWRL/