2014 6th International Conference on Cyber Conflict
P.Brangetto, M.Maybaum, J.Stinissen (Eds.)
2014 © NATO CCD COE Publications, Tallinn

Situational awareness and information collection from critical infrastructure

Jussi Timonen
Department of Military Technology
The Finnish Defence Forces
Helsinki, Finland
jussi.timonen@mil.fi

Samir Puuska
Department of Computer Science
University of Helsinki
Helsinki, Finland
puuska@cs.helsinki.fi

Lauri Lääperi
Department of Military Technology
The Finnish Defence Forces
Helsinki, Finland
lauri.laaperi@mil.fi

Jouko Vankka
Department of Military Technology
The Finnish Defence Forces
Helsinki, Finland
Jouko.vankka@mil.fi

Lauri Rummukainen
Department of Military Technology
The Finnish Defence Forces
Helsinki, Finland
lauri.rummukainen@mil.fi

Abstract: Critical infrastructure (CI) is a complex part of society consisting of multiple sectors. Although these sectors are usually administered independently, they are functionally interconnected and interdependent. This paper presents a concept and a system that is able to provide the common operating picture (COP) of critical infrastructure (CI). The goal is to provide support for decision making on different management layers. The developed Situational Awareness of Critical Infrastructure and Networks (SACIN) framework implements key features of the system and is used to evaluate the concept.

The architecture for the SACIN framework combines an agent-based brokered architecture and Joint Directors of Laboratories (JDL) data fusion model. In the SACIN context, agent software produces events from the source systems and is maintained by the source system expert. The expert plays an important role, as he or she is the specialist in understanding the source system. He or she determines the meaningful events from the system with provided guidelines. The brokered architecture provides scalable platform to allow a large number of software agents and multiple analysis components to collaborate, in accordance with the JDL model. A modular and scalable user interface is provided through a web application and is usable for all SACIN participants. One of the main incentives for actors to provide data to the SACIN is the resultant access to the created COP.

The proposed concept provides improved situational awareness by modeling the complex dependency network within CI. The current state of the infrastructure can be determined by combining and analyzing event streams. Future states can be proactively determined by modeling dependencies between actors. Additionally, it is possible to evaluate the impact of an event by simulating different scenarios according to real-world and hypothetical use cases. As a result, understanding of CI and the ability to react to anomalies is improved amongst the decision makers.

Keywords: *Common Operating Picture, Critical Infrastructure, Situational Awareness, JDL data fusion model*

1. INTRODUCTION

This research presents the Situational Awareness of Critical infrastructure and Networks (SACIN) framework for gathering information from the different entities of critical infrastructure (CI). The main contributions of this paper are the created concept framework and the designed SACIN framework, including the implemented demonstration system. The framework provides tools for gathering information from CI, architecture for information fusion, and a user interface. Based on the derived information, it is possible to support decision making and expand the scope from situational awareness to a decision-making platform.

CI consists of a large number of different and constantly evolving source systems, which are impossible to integrate directly together. A big data system, where raw data from the source systems is gathered and analyzed, is not feasible in this context, because no single entity can understand the operation of all CI sectors. Additionally, most CI systems are privately administered and use equipment to which vendors are not usually allowing access. The solution for the system in this kind of environment is agent-based architecture, where some responsibility of the data integration is placed on the source system experts. The agent is a tool that is able to produce events from the system being monitored and to deliver them onwards. The autonomous agent enables information to be gathered from the source system without affecting the system being monitored.

Our approach is technical; first, we define the problem to be solved in chapter 1 and explore the prior research in chapter 2. In chapter 3, the concept framework is presented, and the architecture supporting the framework is studied in chapter 4. The designed agent component is presented in chapter 5 and the user interface in chapter 6. The empirical part of the study is the implementation discussed throughout chapters 3–6. Finally, in chapter 7, the results and future research are discussed.

2. RELATED WORK

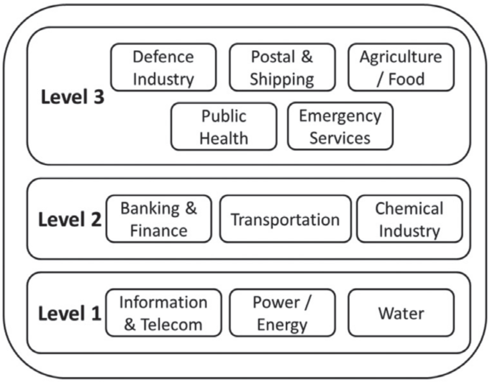
A basic information source on CI protection is the book by Lewis [1]. It presents a model of the sectors in CI and evaluates the threats faced. Modeling CI for use in different simulations is evaluated by Tolone et al. [2]. A project worth mentioning is the Executive order 13636 – Preliminary Cybersecurity Framework [3]. This order presents the basics for a risk based framework; its purpose is to unify and provide an improved understanding of the situation inside organizations. Wide-area situational awareness methodological framework is presented by Alcaraz and Lopez [4]. This research focuses on improving the situational awareness of CIs. An agent-based solution for modeling and simulation of interdependencies in the CI is presented by Casalicchio [5]. This study presents Agent-based Modelling and Simulation Framework, which is also implemented and tested. Attwood et al. present the Smart Cities Critical Infrastructure Response Framework [6]. This framework aims to provide an understanding of linked infrastructure and enable more efficient reactions on failing entities. The dependencies in CI are analyzed [7-10].

According to the literature review, there seems to be a lack of applying the Joint Directories of Laboratories (JDL) data fusion model with an agent-based solution to CI protection. Therefore, this paper combines these two approaches for the use of the common operating picture (COP) of CI. Additionally, the paper presents a concept framework that includes an implementation of the designed system.

3. CONCEPT FRAMEWORK

An important basis for the study is the taxonomy of CI defined by Lewis [1]. This taxonomy, presented in Figure 1, operates as a guide for dividing the entities in CI. Furthermore, the taxonomy provides a means to understand the interdependencies of objects in CI. The taxonomy is applied throughout the framework from low-level components to the COP. The taxonomy is complemented with event ratings [11] and event categories [12].

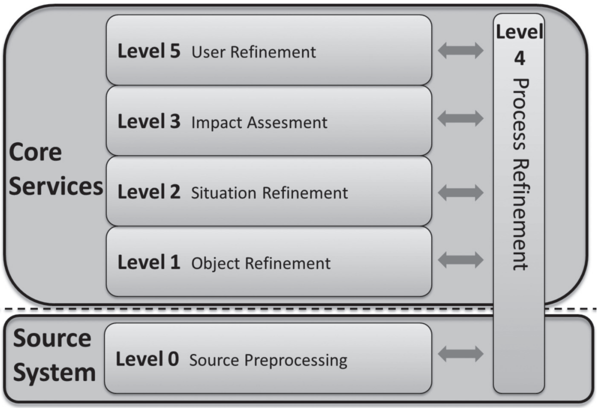
FIGURE 1 – SECTORS OF CI [1]



The vast amount of dependencies has an important role in the concept framework. The strength is in understanding the dependencies amongst the systems. For this purpose, the concept includes means to define and analyze the dependencies. The goal is to offer a source system in the CI means to share information and update the relations to the other entities.

The data fusion model used for SACIN is the JDL model, which presents a process supporting data collection and integration for the COP. The implementation of JDL model to cyberspace has been studied in [13, 14]. Challenges of information and data fusion in the context of urban operations are examined in [15-17]. Although the applied environment differs [15-17], the challenges in fusion are remarkably similar. The JDL model applied to SACIN is presented in Figure 2.

FIGURE 2 – JDL ADOPTED FROM [18]

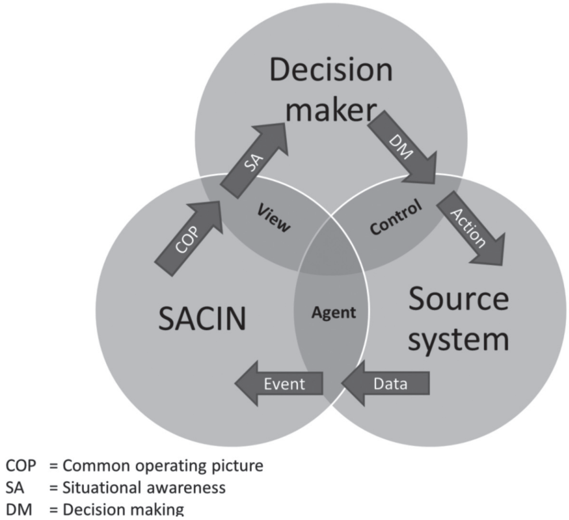

Level 5 User Refinement
Level 3 Impact Assesment
Core Services
Level 2 Situation Refinement
Level 1 Object Refinement
Source System
Level 0 Source Preprocessing
Level 4 Process Refinement

CI source systems with their monitoring components act as a sensor in the fusion process point of view. These systems are integrated with the SACIN through the agent software belonging to the JDL level 0. The purpose of the first, second, and third fusion levels is to analyze and form a model of CI in its current and future state. Analysis is initiated at level 1 by creating objects from the event stream. Objects can be created from just one significant event or from information gained through multiple events. For example, recognizing systematic port scans from multiple agents could create more serious reconnaissance objects. The aim for level 2 is to combine the information from the objects delivered from level 1 and construct the current state of the whole system. The acquired system state is then supplemented with the information at level 3. The focus on level 3 is the prediction the futures risks, possible vulnerabilities, and an estimation of their effects. Level 4 provides the ability for the system to control its operation through automated and user defined mechanisms. Finally, the COP is presented to the user with the level 5 user interface.

In Figure 3, the action flow in the concept framework is presented. The process starts from the source system, which provides data to the SACIN. Data collection is made possible by creating an agent component, which can be deployed directly to the source system. The agent is able to connect to the SACIN and also extract important data from the source system. From these data, the agent produces events that are directed to the SACIN framework. From these events, the SACIN creates the COP according to the JDL model (Figure 2). The user interface supports the situational awareness of the decision makers at different levels.

In Figure 3, the decision maker includes authorities and source system operators. Authorities focus on maintaining society, whereas source system operators provide data to the SACIN and aim to improve their own processes. In the context of this study, a straight gateway for the means of effect (MoE) is not offered to the authorities' level, since the source systems are usually not owned or controlled by the high level decision makers. Control and action represent the communication between source system operators and authorities via every possible gateway (automated, email, phone, etc.).

FIGURE 3 – RELATIONS OF THE ENTITIES



An important part of the source system is the domain expert, who is responsible for understanding the state of the particular source system. The agents deployed to the source systems create and deliver the data forward. These data are aggregated using operators (human) and analyzers (automatic) to detect the relationships between the events based on dependencies, adding information from external systems, developing conclusions, and combining information. The COP is being created by SACIN, and the resolution is maintained all the way to the individual source systems and complemented with the aggregated information and dependencies. Source system-specific views are available amongst all actors. Furthermore, the COP is available in its entirety to the authorities. From this information, it is possible for a decision maker at the authority level to supplement one's situational awareness and use the desired MoEs. In Table 1, the different roles in decision making are presented.

FIGURE 4 – CREATION OF THE COP OF CI

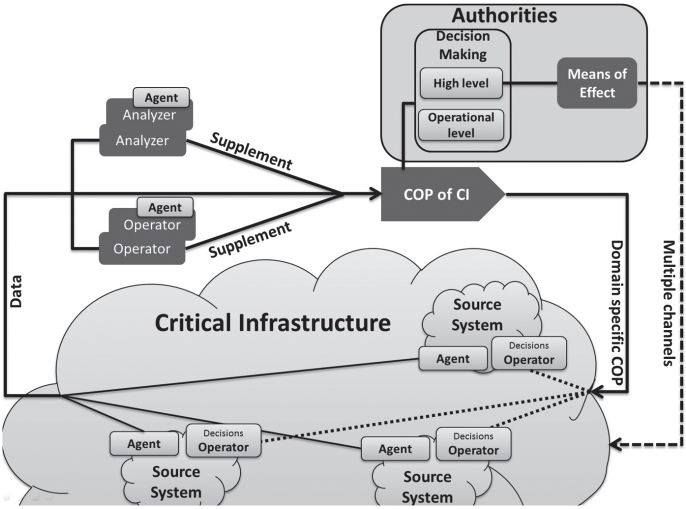
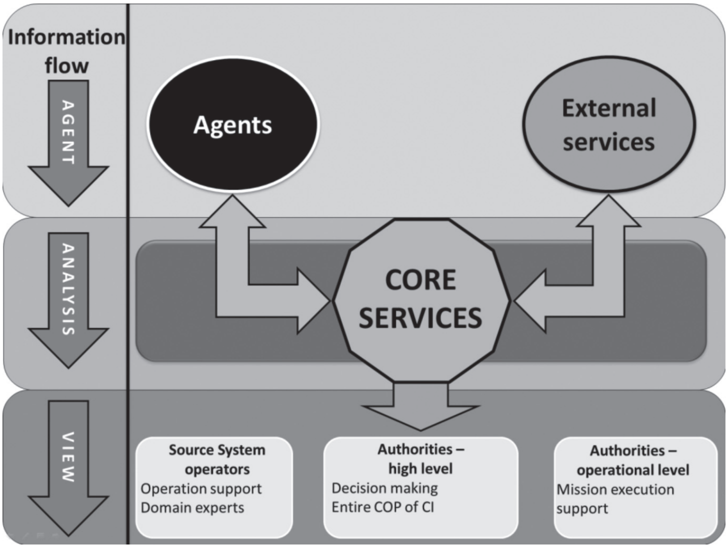


TABLE 1 – DECISION MAKING

Actor	Decisions	Means of Effect	Purpose for using SACIN	Examples of the entities
Source System operator	-Decisions effecting one's own system -Business-related decision.	-Internal means of the source system -Control over the own system	-Improved SA of the surroundings and connecting entities -Prediction and improving resilience of one's own business	Power grid operator, water supply company
Authorities high level	-Decisions effecting society as a whole -How to deal with and recover from a situation of crisis -Prediction and simulation of complex event chains	-Political -Military -Information sharing -Guidance -Preparation (emergency supply plans)	-To protect society from crisis situations -Improved recovery -To test the scenarios	Ministries, council of the state
Authorities operational level	-Decisions concerning one's own operations	-One's own operations	-To improve the efficiency and predictability of one's own operations	Police, fire department, rescue department

A SACIN core service (Figure 5) is the component where information is analyzed, stored, and organized. The agents are connected to the core services using a two-way communication channel. The final layer from the perspective of information flow is the view, where the analyzed information is delivered from the core services to the user interface. The operators of the user interface are fundamentally the same as presented in Table 1. Information providers (source systems) are interested in the state of CI on which they are dependent.

FIGURE 5 - CONCEPT FRAMEWORK



4. SYSTEM ARCHITECHTURE

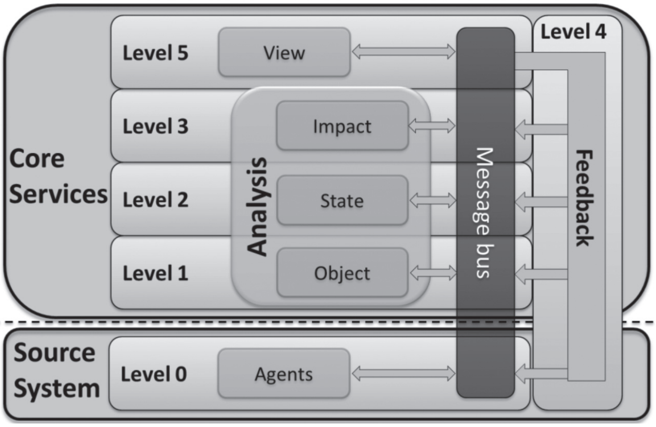
The main goal of the SACIN system architecture is to provide a platform supporting data integration and analysis of CI sectors. Different JDL data fusion processes should be supported to allow the integration of different CI systems. Scalability, closed systems, and data privacy are only a few requirements that lead to an agent-based integration approach. From the architecture point of view, the JDL model (Figure 2) and agent-based approach are the main influences regarding critical design choices.

The JDL model itself does not take a stand on architectural decisions; it defines required steps the system must be able to offer. The architecture must accommodate all six data fusion sub processes and allow them to work together in a flexible and scalable way. The inter component communication channel is the key feature allowing operation in distributed environments and implementation on a national scale. Sufficient communication channels can be achieved with a common message bus.

Requirements for the common message bus are first, to have the capacity to handle large numbers of messages from multiple sources and second, to allow the routing of message to one or multiple destinations. The first requirement is to allow a large number of agents to send information from their respective systems to analyzer components. The second one allows a flexible and scalable way for a component to communicate with any other one within the fusion chain. The role of the message bus is central to the functioning of the system. Therefore, it is important to be able to scale the capacity by distributing the load to multiple servers as well as to ensure service availability by duplicating the access points.

Figure 6 depicts a logical architecture diagram for the SACIN following the JDL model. All fusion sub processes are handled with respective components that communicate through the common message bus. Separation between domain and SACIN entities presents the administrative boundary between systems. The agent acts as a middle component between the separately administered source system and the SACIN. Event analysis is separated into three different components, which together, provide current and future states of CI. The analysis result is presented to the users through the view component in the form of the COP.

FIGURE 6 – JDL AND ARCHITECTURE



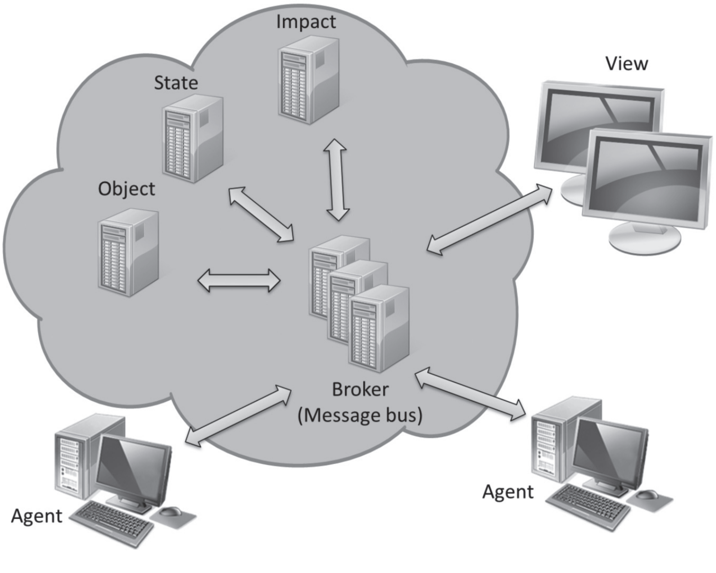
The message bus functionality can be achieved by various technologies, such as an enterprise service bus, a p2p network, or as a cloud service. The most important function of the message bus is to allow a large number of agents to send their events to the analyzers. Additionally, there may be separate analysis components that require the same streams through broadcasting. It is necessary to keep the system simple to manage, and the agent in particular should be able to run on low-end equipment.

A suitable technology implementing the message bus is brokered architecture from a cloud service point of view. The broker can be seen as a cloud service where a group of servers together offer message transfer services. Various services such as broadcasting and bi-directional messaging can be offered with little overhead. The same events can be directed to multiple destinations almost simultaneously. Additionally, as most of the communication between components is the “fire and forget” type, brokers can easily allow all inter component communication.

Although data fusion is the system's primary task, there are a few other topics that need to be addressed before the system is functional. The first and most important one is agent identification, which is required for separating and linking events to source systems. Because a large number of agents may be present, the ID pace should be large. Additionally, ids should be allocated randomly to make it more challenging to enlist brute force or guess used IDs. The second is the handling of user accounts that are used to operate within the system. User accounts are necessary for assigning ownership status to the agents. There needs to be a registrar component that is responsible for allocating and registering the agent ids as well as users to the system.

Figure 7 presents the interactions between different components. The broker acts as an intermediate service for routing messages between components. It does not orchestrate the operation in any way, but only allows inter component communication. All the operation logic and actions originate from the components and users. The broker, i.e., message bus, and other presented components together form a SACIN framework, which allows the integration of data from a separate CI sector. SACIN system components should be as separate and independent as possible. Each component should define an interface that other components are able to use through the message bus. Interfaces allow the addition of third-party services in the analysis chain. More sophisticated components complement the basic functionalities provided by SACIN. The primary messaging format between different components is an event. The agent component is presented in more detail in chapter 5 and user interface in chapter 6.

FIGURE 7 – BROKER



Scalability is a major requirement for the common operating picture system as the goal is to allow implementation on national scale. Therefore introducing new information sources, i.e. agents, to the system should increase resource requirements as little as possible. Networking between agents and analysis components should be flexible enough to allow traffic load sharing between multiple servers.

In accordance to the JDL model, the agent is the interface between the source system and the SACIN. The level 0 source pre-processing allows the addition of new source systems that differ considerably from the other ones. Additionally, the agent acts as a low pass filter when it analyses and categorizes the source systems raw data. By reporting only relevant events to the SACIN the amount of transmitted data can be reduced greatly and not to overwhelm the broker servers. On average the expected amount of traffic from agents shouldn't be more than a few events per minute and a few events per second when certain incident occur.

Although core analysis components of the system are affected by the number of agents that produce events to the system they should not be the bottleneck of the system. As the JDL model levels 1 to 3 are all able to continue the filtering of the input data they can limit the traffic volume on such levels that the core services are not congested. Especially level 1 object refinement has an important role as it is the first analysis component handling the events. Although the filtering can reduce the load to other levels the level 1 must support load balancing to multiple servers. As the level 1 analysis focuses more on individual agents than dependencies between agents, it is possible to separate agent to groups that are handled by dedicated servers.

Analysis

As mentioned above, the analysis components produce events at object, state, and impact levels (see Figure 6). These follow the JDL data fusion model and handle the tasks defined in chapter 3. All analysis components are connected through the message bus and therefore can be distributed to separate servers. However, the state analyzers require access to the common database to achieve state for the whole system, and the impact analyzer requires access to the dependency information between different source systems.

Object

The object analyzer is responsible for handling the events that originate from agents. It analyzes the event streams and filters out the desired events. Additionally, object analyzer can detect and generate new events by combining information from different sources. For example, if level one analyser detects multiple port scanning operations in a given time frame which are directed to multiple agents in one sector or geographical location, a new event with greater severity can be generated to represent a possible network reconnaissance. Complex event processing techniques should be utilized in this analysis because the input is event stream [19].

State

The state analyzer forms states of all source systems based on object analysis events. Here the state is linked with agents and stored in the database. State information is constantly updated and new events are generated as the state changes. Additionally, current states of the agents can be queried through the message bus by other components. Severity of the events is largely assessed on the source system experts when they are defining how severely the detected event affects their own system operation.

Impact

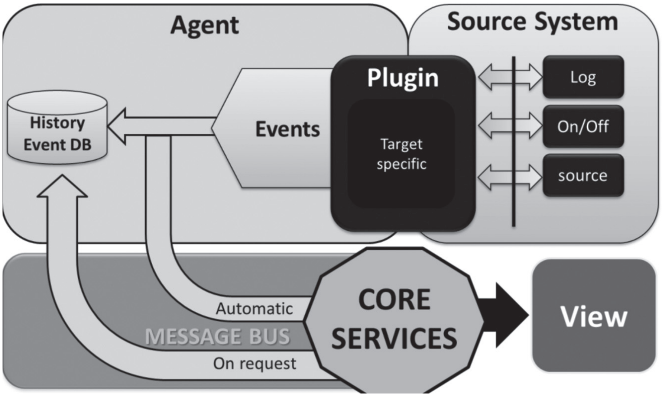
The impact analyzer focuses on determining the future state of the CI. Dependency information between different source systems, i.e., agents, is required for the analysis to allow various network analysis methods to be utilized. For example, vulnerability analysis can be performed to detect critical nodes or failure propagation throughout CI. Additionally, the alarms can be quickly propagated to specific systems to inform incidents such as telecommunication power outages.

5. AGENT

A SACIN agent (see Figures 4 & 7) is a middleware component designed to facilitate centralized event logging and analysis. All agents are assigned unique identifiers, which are used to separate them from each other within the SACIN framework. The purpose is to collect and log events from diverse sources and unify the event format for further analysis. A SACIN agent is designed to collect important status information from systems or processes that are part of CI. These systems can vary from industrial automation to custom intrusion detection systems. Because these systems have vastly different logging and error reporting capabilities, the middleware approach provides the needed flexibility between ease-of-use and wide compatibility.

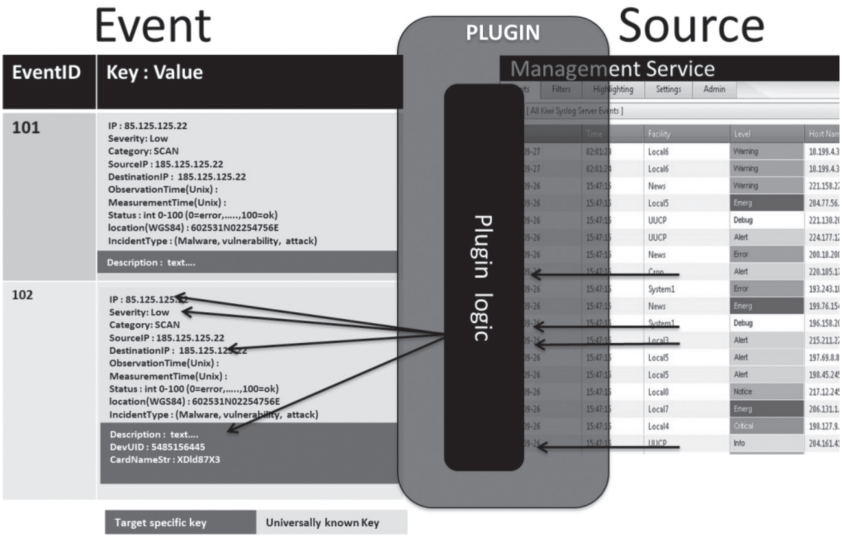
Figure 8 describes the agent attachment to the source system. The actual event generation is done by the SACIN agent through a domain-specific software component called a plugin. This component will be built by a source system expert and it will take care of gaining and interpreting system incidents and providing events to the SACIN. The agent stores the events into a database, if allowed by the used platform, from which it is possible to collect the events for a more detailed analysis of the core services.

FIGURE 8 – AGENT



The most common place for the agent to be installed is a centralized component controlling a large entity from the same branch. The output of the system is used to understand the situation of the source system. Figure 9 presents an example of the information flow when the agent generates events from a log file. From the perspective of SACIN, there is no direct visibility to the actual source system or the cause of event. The source system expert is offered tools and guidelines to be able to build the plugin. This means that the source system expert is, in fact, responsible for making system observations.

FIGURE 9 – PLUGIN



EventID	Key : Value
101	IP : 85.125.125.22 Severity: Low Category: SCAN SourceIP : 185.125.125.22 DestinationIP : 185.125.125.22 ObservationTime(Unix) : MeasurementTime(Unix) : Status : int 0-100 (0=error,.....,100=ok) location(WGS84) : 602531N02254756E IncidentType : (Malware, vulnerability, attack) Description : text....
102	IP : 85.125.125.22 Severity: Low Category: SCAN SourceIP : 185.125.125.22 DestinationIP : 185.125.125.22 ObservationTime(Unix) : MeasurementTime(Unix) : Status : int 0-100 (0=error,.....,100=ok) location(WGS84) : 602531N02254756E IncidentType : (Malware, vulnerability, attack) Description : text.... DevUID : 5485156445 CardNameStr : XDld87X3

	Time	Facility	Level	Host Nam
09-27	02:01:2	Local6	Warning	10.199.4.3
09-27	02:01:2	Local6	Warning	10.199.4.3
09-26	15:47:1	News	Warning	221.158.2
09-26	15:47:1	Local5	Emerg	204.77.56.
09-26	15:47:1	UUCP	Debug	221.130.2
09-26	15:47:1	UUCP	Alert	224.177.1
09-26	15:47:1	News	Error	200.18.20
09-	15:47:1	Cron	Alert	220.105.1
09-26	15:47:1	System1	Error	193.243.1
09-26	15:47:1	News	Emerg	199.76.15
09-2	15:47:1	System1	Debug	196.158.2
09-2	15:47:1	Local3	Alert	215.211.2
09-26	15:47:1	Local5	Alert	197.69.8.8
09-26	15:47:1	Local5	Alert	198.45.24
09-26	15:47:1	Local0	Notice	217.12.24
09-26	15:47:1	Local7	Emerg	206.131.1.
09-26	15:47:1	Local4	Critical	198.127.9.
09-2	15:47:1	UUCP	Info	204.161.4

6. USER INTERFACE

The user interface of SACIN presents a way to visualize the COP. It serves as level 5 of the JDL model and tries to address user refinement issues, such as workload, visual attention, and particularly situational awareness, as presented by Blasch and Plano [20]. The user interface receives events of the infrastructure from the SACIN system back end. When new events are received, the user interface visualizes them into four different views that attempt to increase the situational awareness of operators of the SACIN system. As interpreted at JDL level 5 [20] and defined by Endsley [21], situational awareness is "the perception of the 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." The views of the SACIN user interface attempt to cover all three aspects of this definition: the knowledge about all actors in the CI environment and their current and future states.

A general overview, presented in Figure 10, of the monitored infrastructure is provided to the operator as a quick way to check whether all parts of the infrastructure are working correctly. The operator has the ability to select the actors he wishes to monitor. The actors of the CI are divided into 11 different sectors according to the industry to which they belong. This categorization follows the taxonomy presented by Lewis [1] (see Figure 1) plus one extra sector for actors that do not necessarily belong to any other sector. The current statuses of each sector are then visualized as six-segmented circles, as shown in Figures 10 and 11. These six segments represent the Federal Agency Incident Categories [22].

FIGURE 10 – OVERVIEW

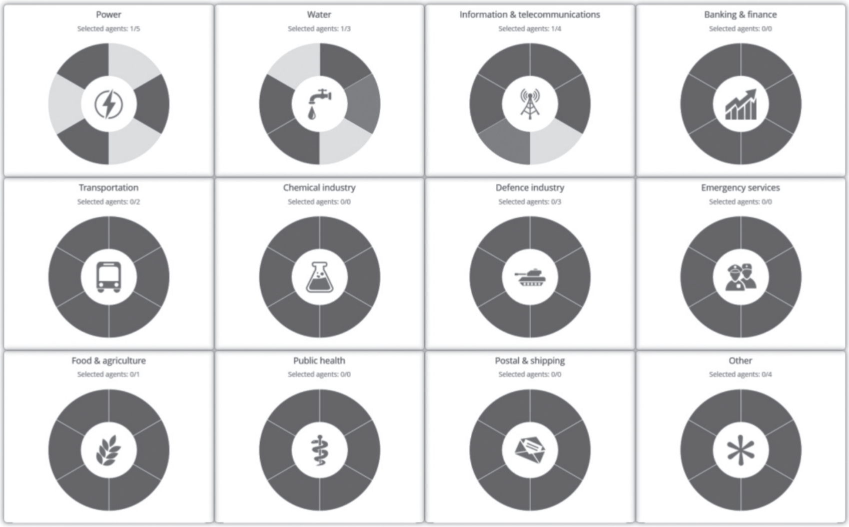
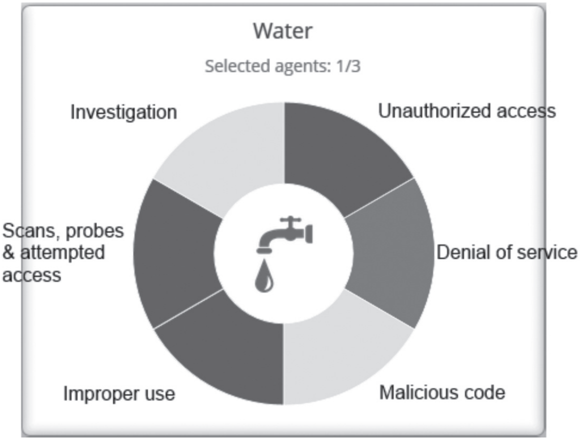
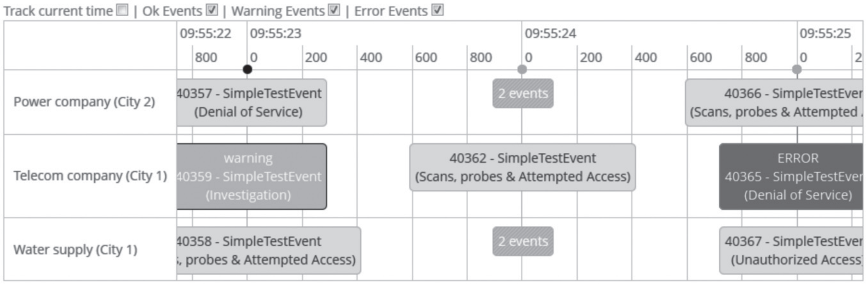


FIGURE 11 – STATUS CIRCLE IN FIGURE 10



A timeline and a common event log, as shown in Figure 12, offer a temporal view for the operator to see when events have actually happened. This way the operator may, for example, analyze consecutive events and link them together even if there are no indications of a relationship between the two in other views or external sources. This offers an advantage when doing risk analysis. Operators also use this view to receipt new events. This ensures that they have consciously seen all the events.

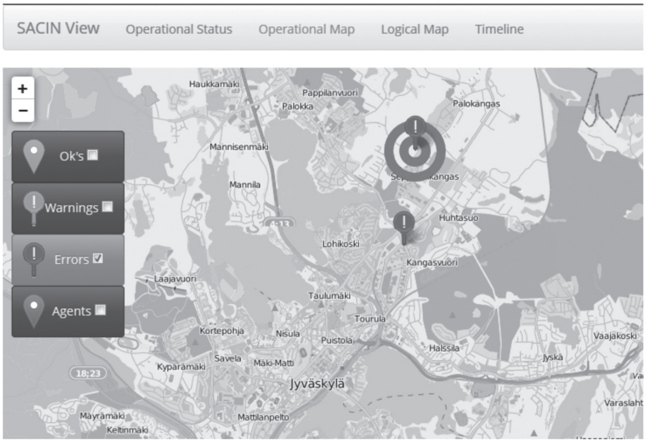
FIGURE 12 – TIMELINE



Track current time □ | Ok Events ☑ | Warning Events ☑ | Error Events ☑

A map view, as shown in Figure 13, is offered to the operator so that the geographical distribution of agents and faults becomes clear. Regional events, such as floods, storms, or alike, may also be spotted on the map view. The implementation itself works as most contemporary map interfaces such as Google Maps. Operators also have the option to filter out types of events in which they are not interested.

FIGURE 13 – MAP AND GEOGRAPHICAL DISTRIBUTION

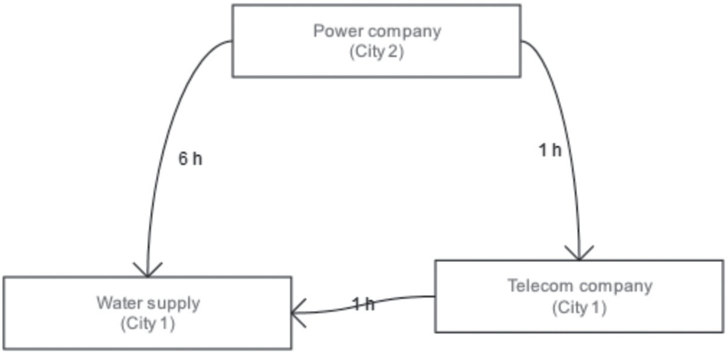


Ok's □
Warnings □
Errors ☑
Agents □

Operators also need knowledge about potential escalating events that may occur. The logical view of the user interface incorporates logical dependencies between different actors in the CI, as shown in Figure 14. For example, a water supply company that is highly dependent on a power station may suffer from system failures due to power outages in the power station.

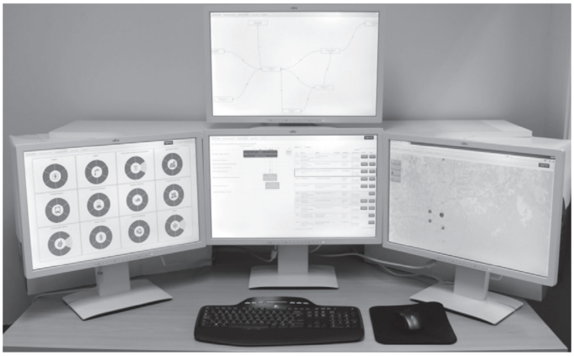
Because of this, an operator at the water supply company needs to know to which actors his company is dependent on and how fast faults will propagate. The dependencies are visualized in a simple directed graph that is drawn based on the selected actors. Each edge is accompanied with a time estimate that tells the operator how long the dependent can function normally without the other actor.

FIGURE 14 – LOGICAL MAP



As stated, these views try to support situational awareness, as operators are shown the current statuses of each industry, offered varied ways to see the events of the CI, and the dependencies between the actors are displayed so a projection of the future is possible. At an operator's workstation, these four views are positioned as shown in Figure 15. The layout is based on the idea that the most interesting view, the timeline, is placed in the center. The overview is placed on the left and the map on the right, so the general workflow supports the left-to-right type of reading. Ideally, an operator first checks the overview to see if everything is working correctly, continues to the timeline view to receipt new events, and finally, examines the map to look for regional events. The logical dependencies view is placed on top of the center view, as it is assumed to be used infrequently.

FIGURE 15 – DISPLAY LAYOUT



The usefulness and performance of the user interface was tested on several test sessions. During these sessions, test participants evaluated the usability of the system using the System Usability Scale (SUS) [23]. It was also tested on how well the system works in a real-life-like simulation using the Situation Awareness Global Assessment Technique (SAGAT) [24]. As a result, the overall SUS score was 71 on average, and all the error events were remembered and placed on a map with an average of 60% hit ratio. Test participants considered the timeline view as the most interesting of all four views. This was backed up by the fact that on average approximately 42 percent of total participant gaze time was focused on the timeline view. The user tests also raised a few issues about the necessary functions in the user interface such as the receipt functionality and the linkage of events between different views.

7. RESULTS AND FUTURE RESEARCH

In this paper, the authors presented a concept framework for creating a COP from CI. The implemented SACIN framework demonstrates the key features of the concept. The main contributions of this paper are the combination of the JDL model and the agent-based architecture, backed up by the implementation. In this paper we also present the results of the user tests carried out to the system operators.

Currently, the functionality corresponding the JDL model levels 0, 1, and 5 is being implemented, while other fusion levels are still in the early stages of development. In other words, events from source systems are created, categorized, rated based on their severity, and transmitted to the user interface. Analysis of the current and future states of the source system has still only been partially implemented.

Future research will focus on analyzing the dependencies and information flow to the system. At this time, SACIN does not implement the module for analysis, but there is an interface to attach the module. Similarly, the user interface will be a subject of further development. The usability tests for this paper were performed at the operator level. In future tests, the decision makers will be included in the testing to a greater extent. This will enable real-world scenario-based operations, as at this point the SACIN has the capability to reflect events from real-world data, in real time or simulated.

REFERENCES:

[1] T. Lewis, *Critical Infrastructure Protection in Homeland Security - Defending a Networked Nation*. Monterey, California: John Wiley & Sons Inc, 2006.

[2] W. Tolone et al., "Critical infrastructure integration modeling and simulation," in *Intelligence and Security Informatics*, Berlin, 2004, pp. 214-225.

[3] The White House, "Executive Order - Improving Critical Infrastructure Cybersecurity," Washington DC, 2013.

[4] C. Alcaraz. and J. Lopez, "Wide-area situational awareness for critical infrastructure protection," in *Computer*, vol. 46, April, 2013, pp. 30-37.

[5] E. Casalicchio et al., "Federated agent-based modeling and simulation approach to study interdependencies in IT critical infrastructures," in *11th IEEE International Symposium Distributed Simulation and Real-Time Applications (DS-RT 2007)*, Greece, 2007, pp. 182-189.

[6] A. Attwood et al., “SCCIR: Smart Cities Critical Infrastructure Response Framework,” *Developments in E-systems Engineering (DeSE)*,United Arab Emirates, 2011, pp. 460-464.

[7] Z. Liu and B. Xi, “COPULA model design and analysis on critical infrastructure interdependency,” in *International Conference on Management Science and Engineering (ICMSE)*, Melbourne, 2012, pp. 1890-1898.

[8] C. Wang et al., “National critical infrastructure modeling and analysis based on complex system theory,” in *First International Conference on Instrumentation, Measurement, Computer, Communication and Control (IMCCC)*, Beijing, 2011, pp. 832-836.

[9] R. Zimmerman, “Decision-making and the vulnerability of interdependent critical infrastructure,” in *International Conference on Systems, Man and Cybernetics*, Hague, 2004, pp. 4059-4063.

[10] R. Zimmerman and C. E. Restrepo, “Analyzing cascading effects within infrastructure sectors for consequence reduction,” in *IEEE Conference on Technologies for Homeland Security (HST ‘09)*, Washington DC, 2009, pp. 165-170.

[11] P. R. Garvey et al., “A macro method for measuring economic-benefit returns on cybersecurity investments: The table top approach,” in *The Journal of International Council on Systems Engineering*, vol. 16, no. 3, December, 2012, pp. 313-328.

[12] C. Stock and P. Curry, “MNE7 Collaborative Cyber Situational Awareness (CCSA) Information Sharing Framework,” 2013.

[13] S. Schreiber-Ehle and W. Koch, “The JDL model of data fusion applied to cyber-defence,” in *Workshop on Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, Bonn, 2012, pp. 116-119.

[14] G. P. Tadda, “Measuring performance of Cyber situation awareness systems,” in *11th International Conference on Information Fusion*, Köln, 2008, pp. 1-8.

[15] M. Bjorkbom, et al., “Localization services for online common operational picture and situation awareness,” *IEEE Access*, vol. 1, November, 2013, pp. 742-757.

[16] J. Timonen and J. Vankka, “Enhancing situational awareness by means of visualization and information integration of sensor networks,” in *Proc. SPIE 8756, Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications*, Baltimore, 2013.

[17] R. Virrankoski, “Wireless sensor systems in indoor situation modeling II (WISM II),” Proceedings of the University of Vaasa, Vaasa, Tech. Rep. 2013.

[18] N. A. Giacobe, “Application of the JDL data fusion process model for cyber security,” in *Proc. SPIE 7710 Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications*, Orlando, 2010.

[19] S. Vranes, et al., “Application of Complex Event Processing Paradigm in Situation Awareness and Management,” *22nd International Workshop on Database and Expert Systems Applications*, Tolouse, 2011, pp. 289-293.

[20] E.P. Blasch and S. Plano, “JDL level 5 fusion model: User refinement issues and applications in group tracking,” in *Proc. SPIE 4729, Aerosense*, 2002, pp. 270-279.

[21] M.R. Endsley, “Toward a theory of situation awareness in dynamic systems,” *Human Factors*, vol. 37, no. 1, pp. 32-64, March, 1995.

[22] United States Computer Emergency Readiness Team (n.d.). Federal Incident Reporting Guidelines [Online]. Available: https://www.us-cert.gov/government-users/reporting-requirements

[23] J. Brooke, “SUS-A: A quick and dirty usability scale,” in *Usability Evaluation in Industry*, London, United Kingdom: Taylor & Francis, 1996, pp. 189-194.

[24] M.R. Endsley, “Situation awareness global assessment technique (SAGAT),” *Aerospace and Electronic Conference (NAECON)*, vol. 3, pp. 789-795, 1988.