Situation Awareness and Decision Making for Constituent Systems

Pontus Svenson and Jakob Axelsson
RISE Research Institutes of Sweden
pontus.svenson@ri.se, jakob.axelsson@ri.se

***Abstract*—** The constituent systems (CS) that together form a system of systems (SoS) have a continuous need to assess situations and make decisions. In addition to operating in the environment, they also need to decide upon their status in the system of systems and be aware of their relations to constellations in the system of systems. To be able to make the best possible decisions, the constituent systems need to have an accurate situation awareness, *i.e.,* an understanding of the environment they are in, what other elements are present therein and how this will develop in the future. In this paper we present an analysis of the situation awareness needs of a constituent system in different stages of its life-cycle: ignorant of the SoS; prepared for joining the SoS; a passive CS; an active CS. A conclusion is that CS need to have world models that include information about other objects. Examples from the truck platooning application are given.

Keywords—collaborative system-of-systems, situation awareness, OODA loop, information needs

I. Introduction

Thanks to the advances of computing power and communication system, our society is undergoing a rapid digital transformation. More and more, systems that previously were operated independently are starting to interact and become more and more synchronized. At the same time, the advances in artificial intelligence enable the systems to become more and more autonomous. All of this gives rise to increased benefits for users, and also enables new business use cases. In cases where the systems are, fully or partially, autonomous, they have a need to be aware of their surroundings – they need to have situation awareness. These interacting systems can be referred to as systems of systems (SoS).

In this paper, we introduce the concept of situation awareness for constituent systems and describe an initial analysis of the different awareness needs for different phases of the CS.

A. Systems of systems

SoS are most often defined as independently operated and managed systems that are geographically distributed, undergo evolutionary development and display emergent behavior [1]. The individual systems that together compose the system of system are named the constituent systems (CS) [2]. A set of constituent systems that currently operate together is called a constellation in the overall SoS.

Systems of systems are often classified according to the extent to which they are coordinated [3]: a directed SoS is centrally controlled and designed for a specific purpose; an acknowledged SoS also has a shared purpose and a central manager, but CSs are independently operated; in a collaborative SoS, the CS voluntarily agree upon the purposes, and there is no central controller that can enforce collaboration; finally, a virtual SoS does not even have a shared purpose. In this paper, we will only discuss so-called collaborative systems of systems.

The constituent systems (CS) of the system of system operate in an environment that not only contains other constituent systems but also other entities. In order to make the best possible decisions about its status in the SoS, each CS needs to have an accurate situation awareness. The situation awareness could also be called the "world model" of the CS and is a computer representation of the world. In order to be useful, this world model needs to contain not only the current status of the world, but also short-time predictions about the future states of the observed elements. In many cases, the world model also needs to include historical information, both in aggregated form and sometimes as a searchable data base.

B. Example SoS: truck platooning

As a simple example of a system of system, we will use truck platooning. Truck platooning has been introduced as a way of saving fuel [4] by enabling trucks to drive very close together (thus reducing the aerodynamic drag and hence the fuel consumption). In order to be able to do this, the trucks need to be at least partially autonomous: the gap between trucks needs to be so short that it is not possible for a human driver to brake in time to avoid collisions. By using sensors and vehicle-to-vehicle communication, it is possible to implement control systems that handle the braking and acceleration of the trucks. The first platooning systems only include such longitudinal control, but it is also possible to implement lateral control, alleviating the drivers of the need to steer.

In order for the platooning control system to operate, it needs to have a world model or situation awareness of not only the truck immediately ahead, but also other trucks in the platoon and other road users. For instance, it must be able to handle so-called "cut-ins", vehicles that suddenly cut in before the truck. It also needs to make decisions about joining or leaving a platoon, and in some cases whether to wait or take a detour to be able to join a platoon. All of these decisions require situation awareness in the CS.

In addition to this operational control, the truck (*i.e.*, the CS) also needs to be able to join and leave a platoon (*i.e.*, a constellation). A further complication for platooning is that the amount of fuel saved differs depending on the position in the platoon – a leader truck will save less than a follower. It has been argued [5] that in order for trucks to be able to find each other and also for the cost and benefits of platooning to be shared, it is necessary to have mediating services that communicate with all CS in the SoS and help them find each other. These mediating services also need to have situation awareness.

C. Paper overview

The research questions we target in this paper are thus (1) How can the situation awareness needs of a CS be solved; and (2) What are the information needs of a CS in different phases of its life-cycle.

This work was carried out in part within the MAUS project funded by Vinnova, Sweden's Innovation Agency (grant no. 2019-05100).

The rest of this paper is outlined as follows. Next, we present a brief background on situation awareness and decision making. We then proceed to describe situation awareness for a CS and its relations to other CS, followed by a discussion of the world model of a CS. The lifecycle of a CS is then described, and some of the situation awareness needs in different phases delineated.

II. Related Work in Situation Awareness

Situation awareness (SA) is defined [6] as "the perception of environmental elements with respect to time and/or space, the comprehension of their meaning, and the projection of their status after some variable has changed, such as time”. In the literature on situation awareness, it is most often assumed that the situation awareness resides within the mind of a human operator or decision maker. However, the same concepts can also be used to described either the case of computer-assisted decision-making or fully autonomous operation.

Much of the scientific literature on situation awareness focuses on either how to measure situation awareness in humans [7][8] or on how to use data and information fusion to help humans attain situation awareness [9]. For measuring the degree of situation awareness, the SAGAT method (Situation Awareness Global Assessment Technique) is often used. SAGAT is a query process based on information theory.

For intelligent agents, it is vital to have knowledge about the relevant parts of the world. Sometimes, the information that an agent needs for its decision-making is referred to as the situation picture. The situation picture often contains a geographical representation of the world and the objects therein and this is also important for a CS [10]. For human agents, it is sometimes assumed that this, *i.e.,* a map with symbols, is enough for situation awareness. In many cases, however, the most important information for the decision-maker is non-geographical. It is for instance not enough to know what other objects are present, it is also necessary to know what their intentions are, to be able to predict their behavior in the near future. Geographical representations are not always sufficient for this, and hence it is better to refer to situation information.

What situation information is displayed to a human user is situation, task, and role dependent. To avoid information overload, it is important to filter and aggregate/fuse the information [11]. While autonomous agents have greater information processing capabilities, the need to filter the data used remains – there is a need for perception management [12].

The agent uses the situation information available to increase their situation awareness. In many cases, it is not possible for a human to hold all relevant situation information in their head. Instead, computer representations of parts of the needed information are used, and the situation awareness is distributed amongst human and artefacts. At each time-step, the distributed awareness system needs to determine what information to display to the user. In a similar way, the situation awareness of a CS could be distributed amongst itself and other components of the system of systems.

In cases where there are several intelligent agents that collaborate, it is necessary that the situation information they have access to are consistent with each other. Most often, different users do not need to have the same information displayed to each other – in fact, doing so would lead to information overload and hence a decrease in situation awareness.

Systems of systems as a way of helping humans achieve situation awareness has been used by several authors [13] [14]. However, this literature does not take into account the general needs that are common to Constituent Systems of Systems of Systems in different domains.

III. SA and Decision Making for a CS

A CS, like any intelligent agent, needs to observe the world and decide upon its next action based on its understanding of it. Figure 1 illustrates the so-called OODA loop [15] for a CS. The OODA (Observe-Orient-Decide-Act) loop concept was originally introduced in air combat but has since been used to describe decision making in general. It is similar to the sense-plan-act cycle used in early artificial intelligence literature [16].

It must be noted that the simple OODA loop we use here can be extended considerably: each stage in the loop actually consists of several loops in a recursive structure [17]. The OODA loop model prescribes that an intelligent agent first observes the world, then uses the information observed together with its background knowledge to orient itself in the world. After this, it is possible to make a decision (possibly after constructing a world model and simulating the outcomes of several possible actions) on how to act to reach the agent’s objectives. The action is then performed upon the environment, and the consequences of this and the actions of the other agents is observed, and the loop starts over again.

For the platooning example, each truck needs to observe the immediately preceding truck as well as possible other objects. It then needs to orient itself – is there an intruder vehicle cutting in? Is the preceding truck changing speed or steering? After this, the control system of the platooning truck needs to decide its next action: should it steer, accelerate, brake, initiate an emergency braking?

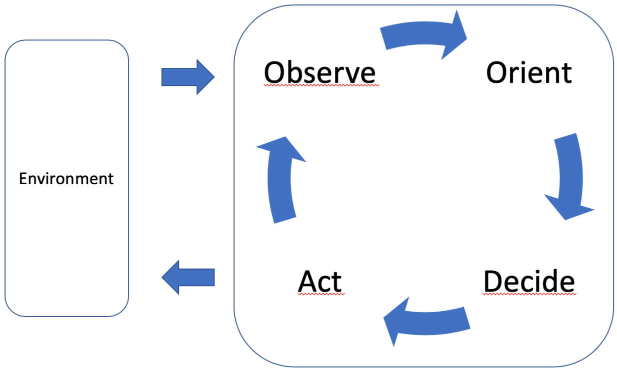

Figure 1 Conceptual illustration of the OODA loop for a constituent system. The CS observes the environment, uses this data together with background knowledge to orient itself (update its situation awareness), then decides on an action which is executed on the environment. In this case, the environment includes both other CS in the same constellation, other CS in the same SoS, and external objects.

Constituent systems of systems of systems have a need to make other decisions than completely independent systems, however. A CS must also decide whether it should be part of a constellation, and if so, which constellation best serves its own goals. Trucks that are not platooning need to look for platooning opportunities, evaluate them and decide whether to join the platoon or not. And once in a platoon, the truck needs to determine if it should leave the platoon.

The constellation also needs to make joint decisions. This could be to allow another CS to join it, or to jointly change behavior – for the platoon it could be to jointly decide upon a braking strategy when the trucks have different braking capabilities. In order to reach a joint decision, the constituent systems of a constellation need to negotiate to determine a compromise that is acceptable to all.

In these decisions, the CS in the constellation need to agree, and any dissenter needs to either leave the constellation or abide by the joint decision. Each CS has its own priorities and goals, and when a joint decision in the constellation is called for these need to be updated and then used to determine the strategy that the CS should use in the negotiation with other CS.

The negotiation with other CS could be modeled using game theory (see [18] for an overview of game theory applications in SoS engineering). Once a result of the negotiation/game has been determined, each CS needs to determine whether it should comply with this decision or leave the constellation. Figure 2 provides an illustration of this process. Note that in most cases, not complying with the joint decision could entail a fine or some other "punishment" as stipulated in the agreements governing the system of system; this needs to be taken into account before the decision.

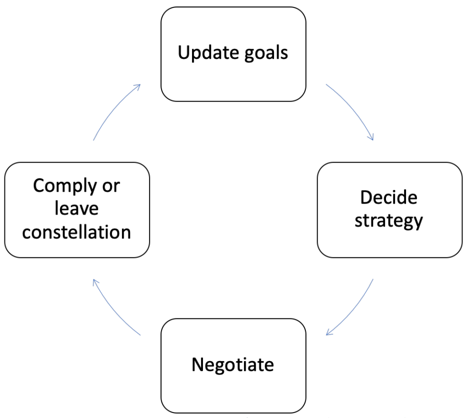

Figure 2 In negotiations, a CS first needs to update its goals, then decide on a strategy. After the negotiation, it needs to determine whether to implement the compromise or leave the constellation.

For platooning, relevant such decisions could be to determine whether the platoon should wait, decrease speed or even make a detour in order to enable another truck to join the platoon. While such operations might be beneficial for the system of systems as a whole, for an individual truck the costs could outweigh the benefits. Consider for example a platoon with four trucks from hauler A and one from hauler B. For the first hauler, as well as for the entire platooning system of systems, it could be beneficial for the platoon to wait for another truck from hauler A. However, the CS from hauler B could choose to leave the platoon since its operating organization does not receive any benefit from the delay.

To avoid individual CS having to make such decisions instantaneously and to ensure consistent behavior, guidelines and rules of operation should be prepared that cover the most commonly occurring cases. While consistent behavior is not a goal in itself for a SoS, it helps to accurate prediction of CS responses, which makes it easier to find good negotiation strategies.

Another example from platooning is how the platoon should act after a cut-in. Recall that a cut-in occurs when an intruder vehicle cuts in between two trucks in a platoon. Different operating organizations could have different rules for when to leave the constellation/platoon in such cases.

IV. World model of a constituent system

In order for a CS to have situation awareness, it must have a world model. A world model is a representation of the real world in a form that the intelligent agent can use to reason about it. It is an important part of the situation information introduced in section III. It can be compared to the sometimes used "digital twin" concept and is the agent's internal representation of its situation awareness. The world model is constructed and updated by making use of all available data sources.

Figure 3 shows an example of the interactions between CS, the environment ("the real world") and mediator services. Each CS constructs its own world model based on its own sensing devices and information received from other CS and mediator services. In addition to the mediator services, each CS is also in communication with its operating organization as well as any other regulatory agencies that are required.

In order for the CS to be able to exchange information with other actors, it is necessary for them to be interoperable [19]. A key concept here is semantic interoperability, which can be defined as the ability of two information systems to communicate with shared understanding (see [20] for a discussion of the relationship of this to fusion).

For platooning, this would correspond to trucks communicating with other trucks (within or outside the platoon) as well as other road users, and their operating organizations. The mediator would be responsible for matchmaking and cost/benefit distribution in the platoon [4].

Figure 4 shows how a CS that uses a fusion engine [11] to update its world model based on data and information received from the environment as well as other CS. The decision agent in the CS (here shown as a human but could be a fully or partially autonomous agent) holds the situation awareness based on the world model and controls the parameters of the fusion engine. A data base of policies and regulations that the CS must adhere to could also be included; note that some rules and regulations are also part of the world model, for instance those that might change depending on the geographical position of the CS.

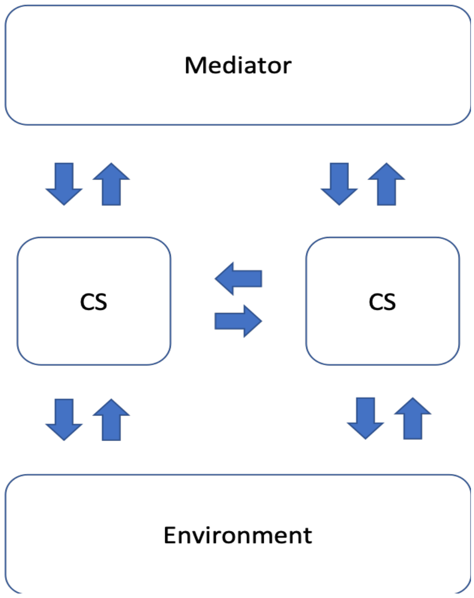

Figure 3 Illustration of the communication and sensing. Each CS communicates with other CS, mediator services, and senses and acts upon the environment.

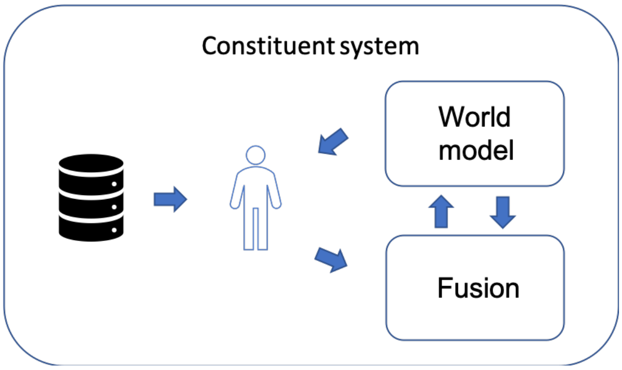

Figure 4 Illustration of the situation awareness components within a constituent system.

For platooning, the world model needs to contain at least information about the preceding truck in the platoon and about any intruder vehicles.

A complicating factor is that a CS can be part of several SoS at the same time. This is illustrated in Figure 5, and also means that the world model needs to be adapted to include all relevant information for all SoS that the CS participates in.

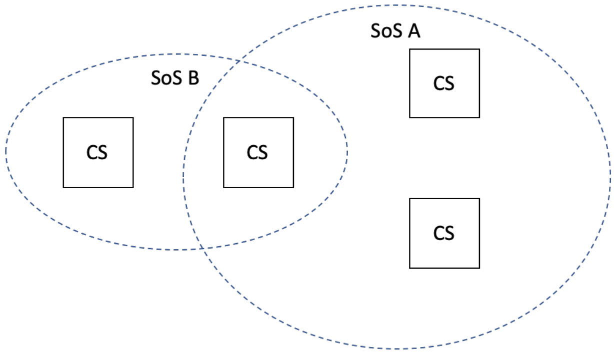

Figure 5 Illustration of how a CS can be part of several SoS at the same time, belong to different constellations in the different SoS.

For platooning, such other SoS that a truck belongs to simultaneously with the platooning system of systems could include the operating organization's own network, the OEM's data sharing and maintenance system, etc.

In addition to the information about the current state of the environment, the world model also needs to include predictions of the future and predictions of the internal states of other CS. Figure 6 illustrates this. Two CS are shown, each of which has its world model (the balloon). Each balloon contains a representation of the entire world – including both CS and their world models. This leads to a recursive inclusion of information about "what the other knows that I know that the other knows that I know…". In the case where all CS are collaborating and voluntarily sharing all information, there is no need for this recursive reasoning. However, most realistic applications will contain CS that do not wish to share all information, *e.g.*, competing companies.

An example of the need for this recursive reasoning in platooning is the need for CS to determine strategies for joining constellations – in evaluating the different opportunities available, they need to consider other CS actions, and those in turn will depend on the first CS actions. Generally, in traffic situations there is always a need to consider what other road users will do, and it is not possible to assume that all agents will share such information.

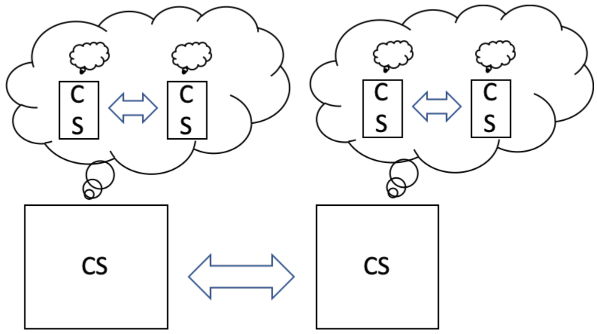

Figure 6 Illustration of the need for recursive situation awareness. Each CS must maintain a world model that includes itself and other relevant CS. This model must include a model of what the other relevant CS know.

V. Information needs during the Life-cycle of a constituent system

To have an accurate world model and obtain situation awareness, it is necessary to know what information needs to be represented in the model. In this section, we determine the information needs of a CS in different phases of its life-cycle.

A. The life cycle of a CS

Figure 7 shows the life cycle of a CS, according to [2]. The life cycle stages are:

- *Ignorant* CS – the CS has the relevant capabilities to contribute to the SoS, but does not meet the requirements of it;
- *Prepared* CS – the CS has been prepared to that is meets the requirements of the CS (*e.g.*, is has the correct technical equipment), but has not explicitly joined it;
- *Passive* CS – the CS is in the SoS, but is not collaborating with other CS in a constellation;
- *Active* CS – the CS is actively collaborating with other CS in a constellation.

Figure 7 shows these stages and the transitions that can be made between them. For each such transition, the CS needs to be able to determine whether or not it should make the transition. At the same time, it needs to keep track of other CS, constellations and even other SoS. A CS must thus be designed so that it is both possible to obtain the needed information and to process it to achieve the desired level of situation awareness. This necessitates the inclusion of world model functionality in CS, as well as fusion engines that can update the world models.

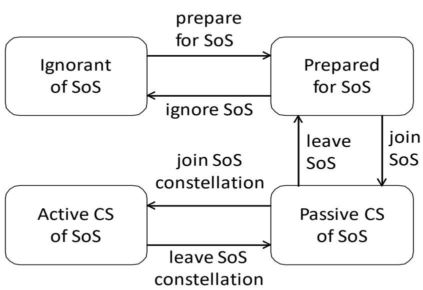

Figure 7 The different phases of a CS and the transitions between them. (Figure from reference [2].)

A first version of the information needs for a CS in the different stages is contained in sub-sections B-E. The method used to obtain these information needs was to construct decision models for the decisions, and finding the root nodes that correspond to needed information. In doing this, we used the platooning example given in [4] as a test case.

B. Ignorant stage

In the ignorant state, the decision to develop the functionality needed to join a SoS has not yet been developed. The managerial organizations (*i.e.*, owning, operating and manufacturing organizations) need to be aware of the:

- Development cost and time needed to prepare for SoS
- Potential benefits of participating in SoS
- Standards and interfaces needed in SoS
- Applicable regulations
- Possible SoS to participate in

C. Prepared state

In the prepared state, the CS has been equipped with the necessary equipment to join a SoS. The operating organization needs to be aware of the:

- Potential SoS to be in (those that are compatible with the installed equipment)
- Benefits and costs associated to each potential SoS
- Applicable Regulations

D. Passive CS stage

In the passive CS stage, the CS has joined a SoS but is not active in a constellation. The CS in this stage needs to be looking for potential collaboration opportunities and could make use of resources and information from its operating organization as well as mediator services for doing this. Its information needs include

- Potential constellations that are reachable and the steps needed to join them
- Existing constellations that are reachable and the steps needed to join them
- The operating and owning organizations of the members of a potential or existing constellation that should be considered, in particular the agreements and regulations applicable to them
- The costs and benefits of joining an existing or a potential constellation
- Potential changes in the environment that would change the benefits/costs of a potential or existing constellation

E. Active CS stage

In the active stage, the CS is cooperating with a constellation. It needs to be in contact with the other members of the constellation as well as its operating organization and mediator services. It needs to have information about

- The current constellation members, their position and other data, operating and owner organization, their future plans
- Other potential and existing constellations as in the passive stage
- Accrued and future costs and benefits of the current constellation
- Updated models for predicting the costs and benefits of remaining in the current constellation or joining any of the others
- Updated models for predicting the possible costs and benefits of leaving the current constellation, either to join another or to continue as a passive CS
- The possible benefits and costs of assuming a different role in the constellation

- Potential changes in environment that would change the benefits/costs of a potential or existing constellation

F. *Additional needs*

Note that the discussion here aims to be general and provide an abstract/general list of situation awareness needs. When analyzing a particular system of system, there will also arise additional case-specific needs. In addition, each autonomously agent needs to have an adequate understanding of its environment in order to, *e.g.,* be able to navigate.

VI. Discussion and future work

We briefly discussed situation awareness for constituent systems of systems of systems. We argued that for the situation awareness needs of a CS to be solved, it is necessary for it to have a world model and a fusion engine. The information that needs to be represented in the world model will differ depending on the current status of the CS, and we provided lists of information needs for different phases of the life-cycle of a CS.

In this paper, we only scratched the surface of subject of situation awareness and systems of systems. Possible future research issues include: further development of the list of information needs; investigations of what information sources and fusion methods should be used to collect that information; exploration of the interoperability needs of constituent systems; analysis of the situation awareness needs of other parts of the system of systems, including mediator services, owner and operating organizations; the effect of uncertainty on the situation awareness in constituent systems; extended analysis of situation awareness needs for negotiating constituent systems, including game theoretical analysis of decision strategies; the interplay between the individual CS, its operating organization, and mediator services in making the decision to join or leave a constellation.

References

[1] M. W. Maier, "Architecting Principles for Systems-of-Systems," INCOSE Int. Symp., vol. 6, no. 1, pp. 565–573, Jul. 1996.

[2] J. Axelsson, "A Refined Terminology on System-of-Systems Substructure and Constituent System States", 14th Annual Systems of Systems Engineering Conference, 2019

[3] J. S. Dahmann and K. J. Baldwin, "Understanding the Current State of US Defense Systems of Systems and the Implications for Systems Engineering," in 2008 2nd Annual IEEE Systems Conference, 2008, pp. 1–7.

[4] J. Axelsson, T. Bergh, A. Johansson, B. Mårdberg, P. Svenson, V. Åkesson, Truck Platooning Business Case Analysis, RISE report 2020:07

[5] J. Axelsson, "Business models and roles for mediating services in a truck platooning system-of-systems", 14th Annual Conference System of Systems Engineering, SoSE, 2019

[6] M. R. Endsley, "The role of situation awareness in naturalistic decision making", In Zsambok, C. E. & G. Klein (Eds.), Naturalistic decision making (pp. 269–283). Mahwah, NJ: LEA, 1997

[7] M. R. Endsley, "Measurement of situation awareness in dynamic systems", Human Factors, 37, 65-84, 1995

[8] M. R. Endsley, "Situation awareness global assessment technique (SAGAT)", Proceedings of the National Aerospace and Electronics Conference (NAECON), 789-795. New York: IEEE, 1988

[9] P. Svenson, R. Forsgren, B. Kylesten, P. Berggren, W. R. Fah, M. S. Choo, J. K. Y Hann, "Swedish-Singapore studies of Bayesian Modelling techniques for tactical Intelligence analysis*",* In Proceedings of the 13th International Conference on Information Fusion (FUSION 2010), Edinburgh, UK, 26−29 July 2010

[10] J. Axelsson, "Needs and Architectural Strategies Related to Geospatial Information in Systems-of-Systems", In. Proc. IEEE Systems-of-Systems Engineering Conf., 2020.

[11] M. E. Liggins, D. L. Hall, J. Llinas, Handbook of Multi-sensor Data Fusion, 2nd edition, CRC Press, 2009

[12] R. Johansson, N. Xiong, "Perception management: an emerging concept for information fusion", Information Fusion vol 3, p 231-234, 2003

[13] P van de Laar, J Tretmans, M Borth (eds), Situation Awareness with Systems of Systems, Springer 2013

[14] J. Brynielsson, A. Horndahl, L. Kaati, C. Mårtenson, P. Svenson, "Development of computerized support tools for intelligence work", Proc. ICCRTS, 2009

[15] J. R. Boyd, Destruction and Creation, U.S. Army Command and General Staff College, 1976

[16] N. J. Nilsson, Artificial Intelligence: A new synthesis, Morgan Kauffman, 1998

[17] B. Brehmer, "The dynamic ooda loop: amalgamating boyd's ooda loop and the cybernetic approach to command and control", Proc. 10th International Command and Control Research and Technology Symposium, 2005

[18] J. Axelsson, "Game theory applications in systems-of-systems engineering: A literature review and synthesis", In 17th Annual Conference on Systems Engineering Research (CSER) - Procedia Computer Science 153, 2019, pp. 154–165.

[19] J. Axelsson, "Achieving System-of-Systems Interoperability Levels Using Linked Data and Ontologies", In Proc. 30th INCOSE International Symposium, 2020.

[20] U. Franke, R. Johansson, C. Mårtenson, P. Svenson, Fusion av information från heterogena källor, FOI-R--3453—SE, 2012