A Risk Modelling Approach for a Communicating System of Systems

Jonathan M. Aitken, Rob Alexander, Tim Kelly
High Integrity Systems Engineering, Department of Computer Science
University of York
York, United Kingdom
Jonathan.Aitken, Robert.Alexander, Tim.Kelly@cs.york.ac.uk

Abstract—**Communicating Systems of Systems provide new methods for developing flexibility and functionality. However, these systems contain added layers of complexity due to the emergent behaviours present in the network. In order to understand the safety implications of operating these SoS this paper develops a new risk modelling technique. Building on information contained within the MODAF model of the SoS this technique provides a structure to enable the analysis of risk.**

Keywords-risk, system of systems, fault tree, hazard, accident, safety

I. INTRODUCTION

Understanding the potential risk in a Systems of Systems (SoS) is important in order to ensure the safety of the operating personnel and civilians within the area. At present there are several different processes that are used within military mission planning that attempt to mitigate risk within an operation. One such technique is Composite Risk Management (CRM) used by the US Army. CRM breaks down a mission into dangerous areas; each component is scored and summed to give a total. However, Johnson [1] questions whether this technique is practicable for personnel to carry out in the field.

The Minimum Equipment List (MEL) is a common method used to define the operability of aircraft. The MEL gives a strict list of permissible flying conditions given a set of inoperable equipment. It does not take into account potential effects of multiple of failures within the aircraft [2]. Henery [2] developed a prototype dynamic MEL, based on the Typhoon safety case. This used a series of fault trees to represent the system. As equipment became unserviceable functionality was removed and the branches organised to represent the new configuration.

We are working on developing a tool for real-time risk awareness in SoS, as discussed in [3].In the next section, we establish a set of definitions that draw a sharp distinction between "SoS safety" and ordinary single-system safety. Following that, we present our risk model and outline a systematic way of building it for a given SoS.

II. GETTING THE TERMINOLOGY RIGHT

If we are going to be successful in SoS safety, we need to be very careful about terminology and definitions. If we define our terms clumsily, then every accident will be "a SoS accident", and this will lead to an enormous expenditure of effort for little gain. We need to prioritise our efforts such that those hazards most deserving on a SoS approach receive one.

Whilst Bibby [4] provides a set of safety related concerns within an SoS there is some ambiguity over the set of constraints included. Some of the conditions included are not accidents, nor direct hazards. Moreover, their scope of concern is extremely wide, and certainly falls outside the traditional scope of safety work. For example, they consider issues such as logistics which can play a part in the severity of an accident, for example by supplying inadequate armour. We believe that the set of definitions in this paper is adequate for SoS safety work, and better in this regard than previous terms.

A. Systems of Systems (SoS)

There are many definitions for the term 'SoS'. For this paper an important one is given by Periorelis and Dobson [5] who state that *"A DSoS (Dependable System of Systems) is a dependable system composed of independent autonomous systems. The purpose of a SoS is to provide a set of enhanced or improved 'emergent' services, based on some or all of the services provided by the participating component systems. The provision of these emergent services requires cooperation between the systems"*.

A SoS consists of systems and personnel. In this paper we refer to both systems and people as *nodes* within the network. The definition of each node is implicit in the design of the SoS. The SoS can be designed using the Ministry of Defence Architectural Framework (MODAF). The individual nodes are grouped together as desired by the designer. The designer takes the decision on how to define an individual node.

B. SoS-Caused Loss

NATO document APP-6A [6] defines the a collection of colours that are used to identify units within a battlespace. These reflect the different possibilities for the affiliation of each unit within the battlespace:

- Blue – friendly unit
- Red – enemy unit
- Yellow – unknown element
- Green – neutral element
- Transparent – unaffiliated or unspecified

This work was sponsored by the UK Ministry of Defence via the Software Systems Engineering Initiative (SSEI)

In this paper, we combine yellow, green and transparent under the umbrella white, i.e. as a unit which is neither red nor blue.

A SoS caused loss is an event that brings about:

- Injury or death to non-combatant or friendly military personnel (blue or white)
- Damage or destruction to friendly military equipment (blue) or other local buildings and possessions (white)
- Damage to friendly military equipment or personnel (blue) caused by undetected-enemy (red) action
- Illegitimate injury or death to enemy (red) combatants

All of the above forms of loss can occur in single system accidents. However, accidents can be described as SoS Accidents if the loss was caused by a SoS Hazard. The prime concern is death or damage caused by the failure of the SoS to act collectively, rather than focusing on single point failure.

C. SoS Accidents and SoS Hazards

1) What is a SoS Accident?

An accident is a "SoS Accident" if and only if it is caused by a SoS Hazard. A SoS Hazard is the combined behaviour of two or more distinct nodes within the SoS that could lead to an accident. An accident that can be described by behaviour confined to a single node (i.e. a single system hazard) is not a SoS accident, even if that node is acting as part of a SoS.

Internal failures (deviations) of nodes *do* need to be explored in SoS hazard and accident analysis. Single-node failure is not enough for an accident to be a SoS Accident; it must be caused by the combined behaviour of two or more nodes.

Many accidents can be superficially described as having been caused by a single system - e.g. simply because they were the last link in a causal chain (that potentially involved other systems). To determine if an accident can be classed as a SoS Accident requires investigation as to whether it happened simply due to "internal" failure of a system, or had contributory behaviour from one or more further systems in the SoS.

2) SoS Accident Classes

BLUE-ON-SELF SOS ACCIDENTS

A Blue-on-Self Accident occurs when action by a single blue unit brings harm on itself through a failure of the SoS to advise against the action. Consider an event where an aircraft flies into a mountain in foggy conditions. It may be that there has been a failure in the navigational systems on board the aircraft. That *is not* a SoS Accident. It *is* a SoS accident if a network node has failed to inform the aircraft about the foggy conditions, which would imply a need for more caution or different route-planning.

BLUE-ON-BLUE SOS ACCIDENTS

This class represents a concept more formally referred to as fratricide. Consider the command to fire a set of projectiles through a busy airspace. If the controller fails to clear the airspace before commencing fire then it may result in a Blue-on-Blue accident. However, if the accident is caused by the projectile not performing to specification (e.g. flying an unexpected course) then it is *not* a SoS Accident as it has not been produced through a failure of collaboration.

BLUE-ON-WHITE SOS ACCIDENTS

The Blue-on-White SoS Accident class represents the possibility of harm to non-combatants in the region. There are several possible scenarios for Blue-on-White SoS Accidents, which are closely linked to Blue-on-Blue – a direct strike on civilians, Collateral damage caused by a nearby strike or collision with civilian vehicles.

RED-ON-BLUE SOS ACCIDENTS

Red-on-Blue accidents are often referred to as "failure to defend" cases, and are not always treated as part of system safety. In our current work, we have included them when they provide interesting examples or introduce special requirements.

BLUE-ON-RED SOS ACCIDENTS

The final class of accidents (Blue-on-Red) refers to illegitimate damage to red personnel or equipment through blue action. This includes the use of excessive force.

3) SoS Hazards

We have already defined a SoS as a collection of nodes collaborating to complete a mission. The SoS Hazards within the network are the cause of the SoS Accidents. We have defined SoS Accidents as being part of four classes. These classes define accidents which have occurred from failures in collaboration within the network. This collaboration is important in managing the operation of the network.

The hazards can arise when there is an error in the collaboration within the network, or a failure of an individual element. This indicates that the combination of states of individual nodes produces the potential for an accident.

The hazards are formed from dangerous state combinations of the individual nodes, potentially combined with certain environmental conditions. This combination of states forms the SoS Hazard. The state combinations play an important role in determining how the network responds. Consider a potential fratricide incident: information on the location and the nature of the target is communicated to an artillery post that carries out action. If the target has been misidentified, and is actually a friendly unit then the situation can lead to a SoS Caused Loss. In this example the internal state of the node has caused a problem. It follows that a clash of the internal states of two or more nodes indicates a hazard within the network caused by the failure of collaboration. We can define the term "SoS Hazard" as: A SoS Hazard is a state of a SoS such that: no further failures are needed for an accident to occur, and it can

be described as a set of state combinations across at least two nodes.

There is no illusion about the difficulties of understanding how node states contribute to accidents. This is especially true with a SoS that displays emergent behaviours. Techniques such as SimHAZAN [7] can provide methods for understanding this link. In SimHAZAN a model of the SoS is created. Multiple software simulations are run with deviations made to the model. Accidents that occur are then recorded with the deviation. The simulation is used to explain how the accident occurred and identify possible hazards. The technique outlined in this paper attempts to build in risk modelling at the design stage that produces risk related information for a SoS during operation.

III. The Risk Model

The risk model consists of a fault tree that can be composed from a mixture of standard and generic (template) elements. The generic structure is shown in Figure 1.

The risk model is divided into several regions, which are described in turn in the following subsections.

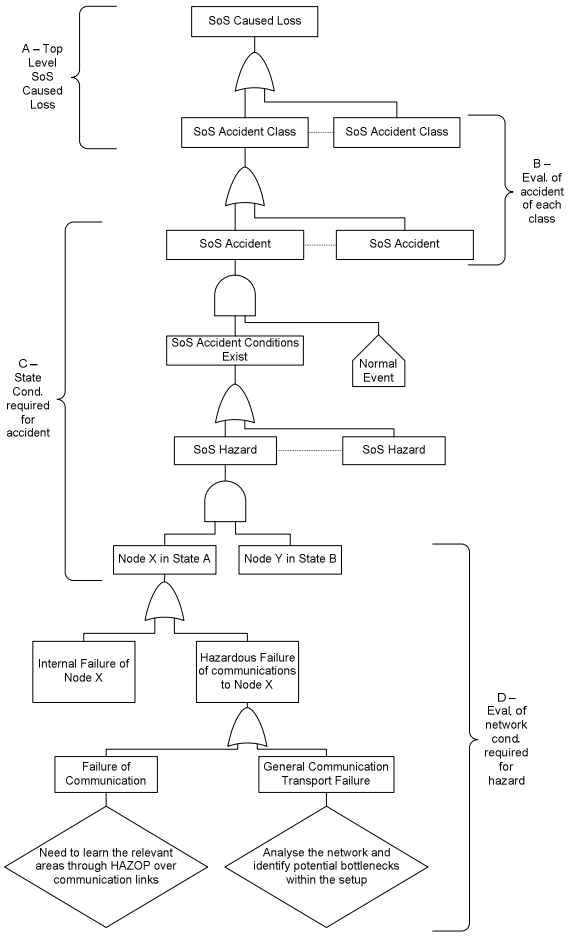

Figure 1 – Basic fault tree for describing a SoS Accident

Traditionally, fault trees are organised so that faults lie in three groups:

- Primary – A component fails for internal reasons.
- Secondary – External excessive stressors, for example environmental temperature effects, cause failure.
- Command – Incorrect control signals cause the component to fail to act correctly.

The definition of an internal node failure (in Region D, Section III.D) comprises both primary and secondary groups. The networked nature of the SoS causes the Command sub-group to be covered by two further groups: generation and transmission.

Each represents a potential command problem – either the information placed on the network is incorrect or conditions within the network cause important information not to be propagated. The level of the investigation for each is subtly different; the first refers to the network and the other to SoS members.

A. Region A

This region brings together the four SoS accident classes described in Section II to the generic category SoS caused loss. The SoS caused loss reflects the top level effects of a SoS accident. By summing across all possible SoS accident classes we can calculate the total risk for each loss type.

B. Region B

Each SoS accident class subsumes various templates (instances) of accidents. These accident templates fall under different classes depending upon the nature of the nodes that are involved. Different accident templates are related to the accident classes in the same way that loss models for aircraft link up concepts to consider the potential for higher level loss. We have developed a number of these templates (see Section IV for an example); there are many more possible.

For example a Blue-on-Blue accident may consist of a friendly fire incident or a collision. In each case a specific Blue-on-White accident can be created from a simple substitution of the nodes involved.

C. Region C

Region C provides a template for the conditions that must be present for a SoS hazard to exist. As part of this region the SoS hazards are linked to their parent SoS accident. This level is built using the definition that the source of SoS hazard is a set of state combinations between at least two nodes – Node X is in State A, Node Y is in State B.

This distinction is important when the template is being filled in with information about the system. It is important to emphasise that we are considering SoS Caused Loss, which by its definition is caused by multiple nodes in different states. Each of the nodes must then be represented as part of the mode. This template encourages the analyst to consider the nodes as a group (at this stage), rather than in isolation.

D. Region D

In the upper regions of the model, we speak in terms of multiple nodes. In Region D, we work with the hazardous behaviour of individual nodes, noting of course that this may, in turn, have been caused by other nodes.

In order to further uncover the cause for the difference the individual state progression must be broken down. The state progression for the node being investigated can be broken down further into two potential categories:

- An internal failure within Node X that has caused an erroneous state change.
- Incorrect external collaboration that feeds back as an input into Node X causing the state change.

A deeper investigation must take place to identify the input failure. Once again there are two clear paths that lead to the Node X receiving inappropriate information.

Firstly any message sent to Node X will travel through the network. Therefore the conditions within the network at the time will affect delivery. Problems with the network will cause corruption and omission or commission errors of the information that can then be explored further. The nature of the network will play a large impact in the success of timely, accurate transfer of information. We are currently developing a network modelling approach that will calculate the risk of these errors.

Secondly a node within the network may generate an incorrect message which is then transmitted to Node X causing the state change. The source node has created incorrect information that has been passed onwards. This definition includes data passed on causing data incest. To cause data incest Node A may forward a piece of information to Node B who passes it to Node C. If Node C returns this information to Node A then the potential for data incest exists. Depending on the context of the message Node A may interpret the returning information as new, providing incorrect affirmation of the original content.

The state change brought about by the incorrect message forms the lowest level of the SoS Accident model. This must be completed by the analyst as part of the risk modelling process.

IV. An Example of an Accident Template

As the fault tree is constructed the SoS Accidents are broken down into SoS Hazards which are split into potential states of the nodes involved. The SoS Accident Templates are used as part of Region C to give areas for expansion in Region D. The construction of each SoS Accident will be different but general templates can be used to provide basic structure.

These templates focus on breaking the accident down into a series of states that caused the accident to arise. Presently we have defined four templates covering a range of potential accidents (only the first of which will be presented here due to space constraints):

- Node X fires on Node Y – The friendly fire case of Blue-on-Blue or Blue-on-White
- Node X fired on by red unit – The failure to defend case for Red-on-Blue
- Node X colliding with Node Y – Another case of Blue-on-Blue or Blue-on-White
- Node X colliding with terrain – A case of Blue-on-Self

A. Node X Fires on Node Y

"Node X Fires on Node Y" covers two of the potential SoS Accident classes, namely Blue-on-Blue and the Blue-on-White. Figure 2 shows the partial fault tree for Node X fires upon Node Y. This encompasses the various different routes that may cause Node X to fire upon Node Y. As more routes are identified this diagram should be expanded.

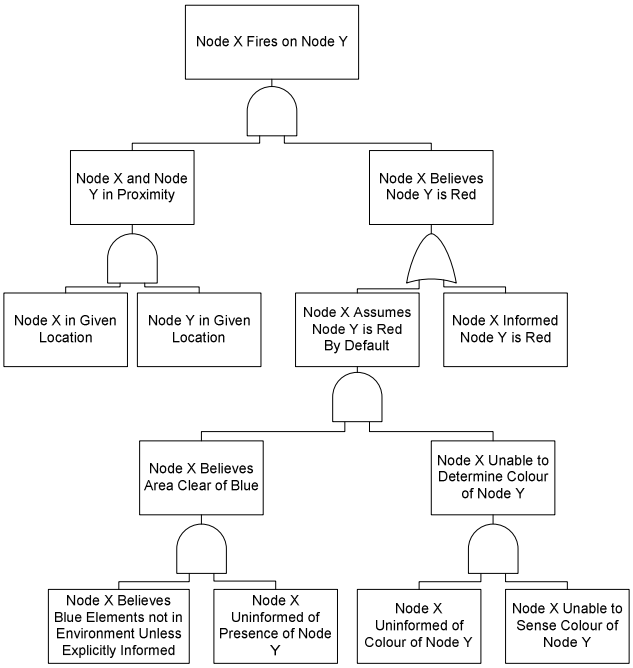

Figure 2 – Partial Fault Tree - Node X Fires on Node Y

Each of these different scenarios involves one node firing at another. In the Blue-on-Blue and Blue-on-White cases this is caused either by a misidentification or failure to recognise Node Y.

This fault tree represents an atomic breakdown of the events that can lead to the decision taken to fire on a unit which is either blue or white. This fault tree is built from two events, that Node X and Node Y are present within range of each other and Node X is believes that Node Y is not blue or white.

The highest left hand region of the fault tree ("Node X and Node Y in Proximity") deals with the Node X and Node Y being in a similar position within the environment at a similar time.

The right hand portion of the fault tree deals starts from the premise that Node X will fire upon Node Y only if it believes that it is an enemy unit. The simplest reason behind this misidentification is that Node X is informed by the SoS that Node Y is red.

B. An Example of a Risk Model

On April 14th 1994 a pair of United States Air Force F-15 Eagle fighter aircraft shot down two United States Army UH-60 Black Hawk helicopters, whilst receiving information that the area was clear from the AWACS aircraft. This led to 26 fatalities [8, 9].

The Black Hawk shoot down provides a useful example of a SoS Caused Loss, and the SoS Risk Model can be applied. Figure 3 shows the top levels of the SoS Risk Model applied to the Black Hawk shoot down.

The top level of the risk model is the SoS Caused Loss. This SoS Caused Loss can be broken down into the separate accident classes, the Blue-on-Self, the Blue-on-Blue, Blue-on-White and Red-On-Blue. In order to focus on the Black Hawk shoot down, only the Blue-on-Blue case will be expanded.

This expansion is carried out using the SoS Accident Templates. This focuses on the Node X fires upon Node Y template that presents the friendly fire case. The F15 aircraft in the network contain an offensive capability. Therefore there are two potential Blue-on-Blue accidents – the F15 may attack either the Black Hawks or the AWACS aircraft. The primary concern in this case is an attack on the Black Hawks; therefore the AWACS case will not be expanded.

In order to fire, the F15 must hold the belief that the Black Hawk is an enemy aircraft. In addition to this, due to the nature of the SoS, the AWACS aircraft must hold the same belief or disagree but fail to communicate it; again for brevity here this will not be expanded.

The SoS Accident Template can then be applied to expand the accident down to the hazards and on to specific components within the SoS. In this case Node X is the F15, Node Y the Black Hawk. Once the bottom levels of the accident template are reached specific analysis on the SoS must take place to expand these elements.

The fault tree in Figure 3 shows a breakdown of the system at the time of the accident. This breakdown has focussed on the interactions of the F15 with the Black Hawks – naturally the AWACS aircraft is involved in the process of misidentification but the primary actors are the Black Hawks and the F15. This fault tree represents a generic template that has been instantiated for the Black Hawk accident.

The information shown in the fault tree is the result of asking simple questions about the interaction between the F15 and the Black Hawks. This analysis could be carried out ahead of time. No guarantees can be made about whether undertaking the analysis before the accident may have helped to avert the accident. However, the process of undertaking the analysis should prompt the analyst to ask questions about the network. For example by breaking down whether the "F15 [is] Unable to Sense Colour of Black Hawk" the analyst should question IFF technology and conclude that a mismatch in codes may bring about this scenario. This then becomes a critical feature that must be ensured when operating in theatre, in the case of this accident IFF codes were not synchronised.

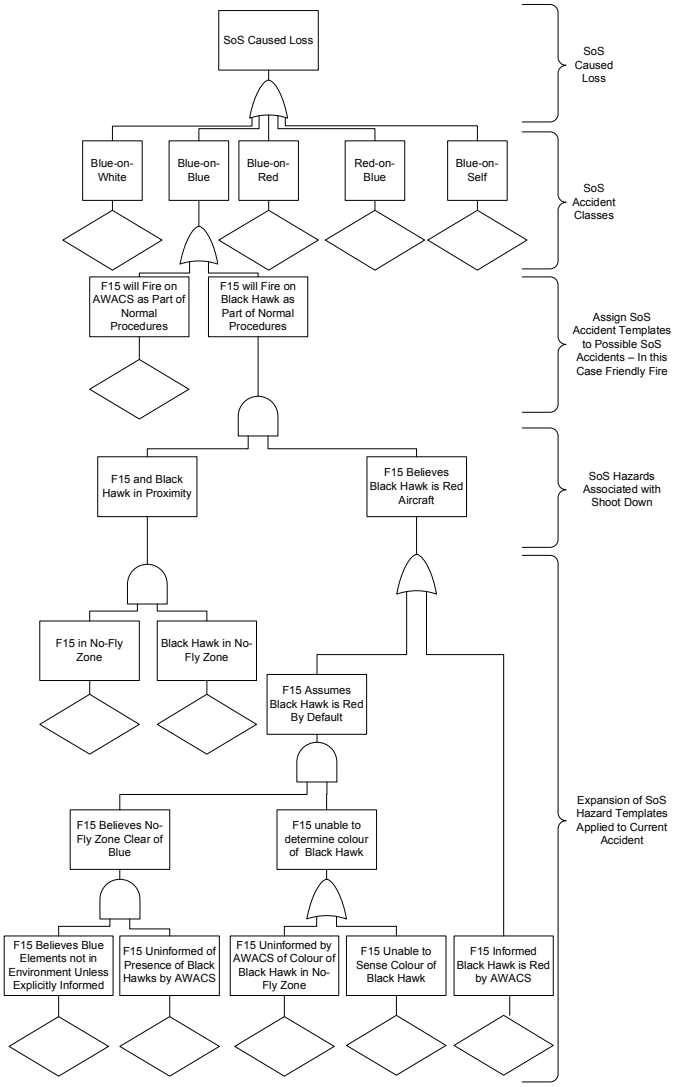

Figure 3 – Analysis of the Black Hawk Shoot Down Using the SoS Risk Model

V. BUILDING A RISK MODEL FOR A SPECIFIC SOS

The process for developing the risk model can be broken down into separate sub-processes for each region. In all cases, we can potentially draw the information we need from a MODAF [10] or DoDAF [11] description of the system. In practice, the quality and suitability of these descriptions will vary; the overall quality of MODAF in practice is something we will seek to assess through practical application, and we will report on this in future publications.

A. Region C

Identify all types of nodes within the SoS. This includes its expected environment, along with white and red nodes that may be in the area. The OV-1 product in MODAF provides a list of the expected blue nodes. Knowledge about red and white

nodes may come from personal experience of the area. Draw up a matrix of pairwise combinations of these, and for each combination identify what classes of accident are plausible.

For each combination of two nodes for which an accident that is plausible, apply one of more accident templates (see Section IV) to find the generic causes of that accident. This template should then be populated with specific information for that SoS, replacing the placeholders (e.g. "Node X") with the specific node names.

B. Region B

This region is simple – classify all the accidents from Region C into the four SoS accident classes given in Section II. Join them together with an OR gate.

C. Region D

This region decomposes the SoS Hazards into the possible causes of the state combination. The model defines two possible scenarios that should be investigated, internal failure and external failure (with the latter being split into network transmission failure and incorrect message contents).

Within MODAF the OV-2 provides a view of the interaction between separate nodes within a SoS and the information that they produce or utilise. These are represented via connections on the OV-2 diagram. This highlights the collaboration between nodes as the mission progresses.

The OV-3 view gives a more detailed picture of the information that is passed between the nodes. The MODAF handbook presents a variety of examples of the use of OV-3 which potentially provides a very rich source of information about the collaborations within the SoS.

Region C is expanded by using the OV-3 to record all of the messages that the node being investigated receives. Each of these messages will have an effect on the node. Some of these changes will be linked to state changes which are available in the OV-6b diagram. Naturally these messages form a chain through the network, from source to sink – the OV-3 includes information about triggering events and accuracy of the data. This provides the opportunity to perform a HAZOP [12] style evaluation of effects caused by messaging failure.

The possibility of internal failure must also be investigated. The SV-4 diagram links each system to the functionality it possesses. The SV-10b shows the usage of the functionality through the state transition process.

D. Region A

This region is simple – connect all the accident classes together using an OR gate.

VI. Analysing a Specific Risk Model

The risk model for SoS accidents takes the form of a standard fault tree. Therefore we can use standard fault tree analysis techniques to calculate risk levels. By annotating the fault tree with the probability of the events occurring then a picture of the overall risk can be built quickly. This can be analysed using ideas based those presented by Henery for the Eurofighter [2]. Probabilities of individual failures are calculated and then propagated up the fault tree, through the different branches to give an overall picture of the risk.

VII. Conclusions and Future Work

Now that we have a risk model, we can look at using it to provide real time analysis of risk, based on the vision we described in [3].

In the future we intend to apply our techniques to a wide category of SoS problems, in both small and real case studies. We hope that others will apply our risk model to their own SoS problems. In addition we will expand our coverage of actual networks and systems to include them as part of the model. Finally this paper defines a risk model for the static configuration of a SoS. We will need to apply the model to dynamic SoS problems to broaden the technique.

References

[1] C. W. Johnson, "The Paradoxes of Military Risk Assessment."

[2] L. Henery, "The Eurofighter 'Operational' Safety Case," Masters Thesis, Department of Computer Science, University of York, 2001.

[3] J. M. Aitken, R. D. Alexander, and T. P. Kelly, "A Case for Dynamic Risk Assessment in NEC Systems of Systems," in 5th International Conference on System of Systems Engineering, Loughborough, UK, 2010.

[4] S. K. Bibby, A. German, P. R. Symons *et al.*, *Resilient and Dependable Systems of Systems*, 2009.

[5] P. Periorelis, and J. Dobson, "Organisational Failures in Dependable Collaborative Enterprise Systems," *Journal of Object Technology,* vol. 1, no. 3, pp. 107-117, 2002.

[6] *APP-6A Military Symbols for Land Based Systems*, NATO, 1999.

[7] R. Alexander, D. Kazakov, and T. Kelly, "System of Systems Hazard Analysis Using Simulation and Machine Learning." pp. 1–14.

[8] S. A. Snook, *Friendly Fire*: Princeton University Press, 2000.

[9] N. G. Leveson, P. Allen, and M.-A. Storey, "The Analysis of a Friendly Fire Accident using a Systems Model of Accidents," in International Conference onf the System Safety Society, Denver, USA, 2002.

[10] "MODAF Website," 2010; http://www.mod.uk/DefenceInternet/AboutDefence/WhatWeDo/InformationManagement/MODAF/.

[11] "DODAF Website," 2010; http://cio-nii.defense.gov/sites/dodaf20/.

[12] *British Standard BS: IEC61882:2002 Hazard and operability studies (HAZOP studies)- Application Guide,* 2002.