SEVENTH FRAMEWORK PROGRAMME

DANSE

Designing for adaptability and evolution in system of systems engineering

Characterization of SoS

D 4.1

Deliverable Information			
Nature	R	**Dissemination Level**	CO
Project	DANSE	**Contract Number**	INFSO-ICT-287716
Deliverable ID	D 4.1	**Date**	2012-08-09
Status	Final	**Version**	1.0
Contact Person	Nir Bendov	**Organisation**	IAI
Phone		**E-Mail**	nbendov@iai.co.il

AUTHORS TABLE

Name	Company	E-Mail
Nir Bendov	IAI	nbendov@iai.co.il
Michael Winokur	IAI	mwinokur@iai.co.il
Tim Lochow	EADS	Tim.lochow@eads.net
Imad Sanduka	EADS	Imad.sanduka@eads.net
Eric Honour	Honourcode, Inc.	ehonour@hcode.com
Stéphane Paul	THALES	stephane.paul@thalesgoup.com
Tim Lochow	EADS	tim.lochow@eads.net
Juergen Mueller	CARMEQ	juergen.mueller@carmeq.com
Riedel, Jan Malte	CARMEQ	jan.malte.riedel@carmeq.com
Roy Kalawsky	LU	r.s.kalawsky@lboro.ac.uk

CHANGE HISTORY

Version	Date	Reason for Change	Pages Affected
0.2	13-Feb-2012	Added Maier SoS definition, current definition chapter moved before partners perspective, added summary chapter.	misc
0.3	17-Jun-2012	Added unedited content in chapter 1, added SoS examples in chapter 3	misc
0.4	22-June	Chapter 3 edits for EADS SoS examples	misc
0.5	28 Jun 2012	Chapter 2 written as draft	misc
0.6	1 Jul 2012	Chapter 1 - introduction - Revised, Chapter 3 - Added Carmeq SoS text, IAI system "A" and "B" edited	misc
0.7	24 Jul 2012	Added IAI SoS E" in chapter and additional edits in chapter 3, added chapter 4	misc
1.0	09 Aug 2012	Formal Release	All

CONTENT

Content of Tables

Content of Figures

1 Introduction

1.1 Overview, Purpose and Scope

The purpose of this document is to review the existing types of SoS and characterize them as the basis for the DANSE new SoS Engineering paradigm.

Although there are many types of SoS surrounding all of us, the SoS of interest to DANSE are principally those which are implemented as an aggregation of pre-existing 'legacy' systems, with added functionality, and new constituent systems developed specifically for the SoS. Consequently this document is not intended as a comprehensive characterization of all possible existing SoS, but only the ones that can be identified of business value to the users in DANSE. To reinforce this point we could, for example, address a very broad span of classification into with social, biological, economic and engineering SoS, but we have elected to concentrate on Engineering SoS that are developed by the industrial users in DANSE.

For that purpose the four industrial partners have made a review of SoS that are developed by them and observe their more relevant characteristics, and not less important the present challenges observed in the development and operation of these SoS.

The material presented in this document will help the project focus on one of its main overarching objectives as stated in the DoW[1]: *DANSE will develop, implement, and test a new methodology to handle the SoS life-cycle, offering evolutionary simulation, analysis, and development of the SoS in real time.*
The DANSE methodology will shape the behavior of an SoS over time as more is learned about the component systems behavior and the emergent behavior due to their interaction, so that the system becomes "correct by evolution." This will be the basis for a new paradigm for SoS Design, development and engineering.
Moreover, researchers working in different areas of SoS applications will be able to use this document as a reference to do similar work in their area of interest and thus help enrich the body of knowledge of SoS Engineering further on.

The document is organized as follows:

Chapter 2 is a summary of Systems of Systems. Current definitions taking from the most up to date sources.

Chapter 3 is the Characterization of SoS from the partner's perspective. It includes a classification schema that was defined as we investigated the existing (and near future) SoS developed and sometimes operated by EADS, IAI, Thales and Carmeq. It also includes a description of some examples that can illustrate to the reader the application of the classification schema we propose.

Chapter 4 reinforces chapter 3 by providing a summary of user challenges that were gathered while asking the experts in the different companies about the different types of SoS. It compliments also the requirements that have been defined for DANSE.

Chapter 5 summarizes the classification schema and the main conclusions we have reached to help us focus the methodology that will be developed in the project

The Appendix brings a very high level overview of the Methodology initial direction of DANSE. The methodology is expected to focus in the class of systems that have been identified in chapter 3 and provide an avenue for the implementation of the DANSE innovations directed at alleviating the user needs.

[1] DoW: Description of Work (Part of the project proposal)

2 Systems of Systems – Current Definitions

2.1 SoS Definitions

Systems of systems were first recognized as distinct entities in the 1990s, as systems and their interactions became increasingly complex. With this increase in complexity, it became apparent to some far-thinking individuals that there was important behavior, of interest to stakeholders that occurred at a level above the constituent systems. With such behavior, it became worthwhile to consider the aggregation of systems into something larger that embodied that behavior. A first idea in this regard was inherent in [Manthorpe 1996], as a definition:

> *A system of systems (SoS) is a system comprised of elements that are systems.*

This understanding has become widespread, but is only partially useful. While it expresses the basic idea, the interpretive definition of the word *system* can apply this definition to nearly any system. Even a microprocessor is comprised of lower-level "systems".

Other authors have tried to make the definition more precise by adding additional thoughts:

> *Systems of systems are large-scale concurrent and distributed systems that are comprised of complex systems.* [Jamshidi 2005]

Jamshidi adds the concepts of "large-scale" and "concurrent and distributed" as applied to the SoS, while also adding the concept of "complex" to the constituent systems. These additional concepts would exclude a microprocessor, but still leave much room for interpretation.

> *A system of systems is defined as a set or arrangement of systems that results when independent and useful systems are integrated into a larger system that delivers unique capabilities.* [US DoD 2006]

The writers of the US DoD *SoSE Guide* add the concept of the emergent behavior ("unique capabilities") of the SoS itself, as well as characterizing the constituent systems as "independent" and "useful." Again, this would exclude the microprocessor because its constituent systems have little use outside the microprocessor – but the microprocessor certainly delivers unique capabilities.

> *A System of Systems is a service and end-user oriented architecture. Systems of systems are currently investigated predominantly in the defence sector. Therefore the most advanced architecture frameworks (e.g. MODAF, NAF, DODAF) originate from the defence sector."* [Schonenborg, ESA 2010]

Schonenborg adds the concepts of service and end-user oriented architectures, implying that the SoS focus is on the higher-level behaviors of the SoS. Jamshidi later described this same characteristic as

> *The system performs and carries out purposes that do not reside in any component system. The principal purposes of the SoS are fulfilled by these behaviours* [Jamshidi 2009]

Yet the microprocessor example also presents a problem for these characteristics as well, because it implements an end-user oriented architecture and also provides higher-level behaviours.

The problem with nearly all such SoS definitions is that they have trouble distinguishing between a *system* and a *system of systems*. The concept of *system* is itself so general as to create this difficulty. Without a clear distinction, the entire concept of *system of systems* suffers from muddy thinking at its outset.

2.2 Maier Characteristics

A solution to the distinction between *system* and *system of systems* was offered by [Maier 1998] in recognition that it is a series of characteristics that make the difference. In addition, these characteristics are not binary. A system becomes more and more like a system of systems as the characteristics become more evident. Maier's characteristics are given as

- **Emergent behavior** – An SoS performs functions that are not achievable by any of the independent component systems. Usually, these functions are only evident at a higher level of perception and do not even make sense at the level of the independent component systems. This particular characteristic is true for both systems and SoS; its presence in the list ensures that the SoS in fact has an existence above and beyond the constituent systems.

- **Geographic distribution** – The constituent systems are distributed in such a way that geographic issues affect the interactions between them – both operational interactions and managerial interactions.
- **Evolutionary development** – Functions and purposes of the SoS are added, removed and modified in an ongoing way. Often, the SoS functions change due to changes in the constituent systems. Such changes, happening asynchronously, result in constant evolution of the SoS.
- **Operational independence** – The constituent systems each have their own independent purposes. If detached from the SoS, the constituent systems are still useful.
- **Managerial independence** – The constituent systems are developed and managed for their own purposes. Relatively unrelated owners with little coordination may perform the management.

The authors of The Open Group Architecture Framework (TOGAF) later acknowledged the same set of characteristics in their description as

> *Systems of Systems should be distinguished from large but monolithic systems by the independence of their elements, their evolutionary nature, emergent behaviours, and a geographic extent that limits the interaction of their elements to information exchange* [TOGAF 2007].

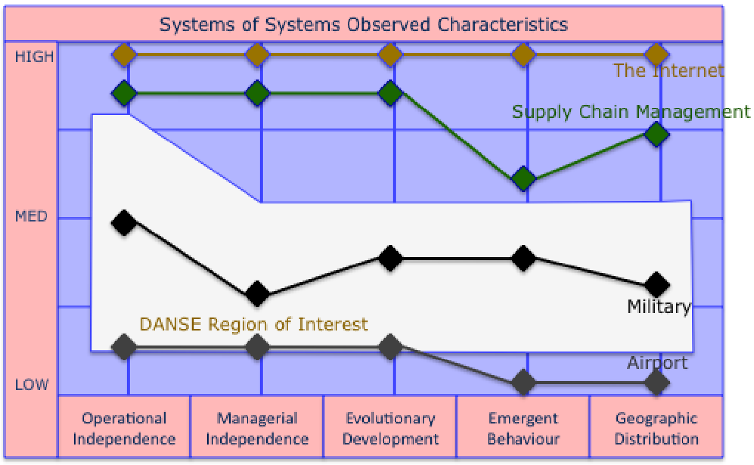

Figure 1 - Interpretive Measurement of "SoS-ness"

These characteristics exist to a greater or lesser extent in each system. The microprocessor example used earlier has emergent behaviour, but has little of the other four characteristics. As systems become larger and more complex, with constituent elements operating as systems in their own rights, the "SoS-ness" of the overall system becomes more apparent. Figure 1 shows subjective measurements of four SoS examples, in which an airport has low SoS characteristics while the Internet is at the top of all five characteristics. The light-colored area represents the SoS characteristics of interest to DANSE.

2.3 SoS Classification

Another way to classify the SoS is also provided by [Maier 1998]. While still recognizing the continuum of measurement in the five characteristics, Maier also found it useful to group SoS into these three major types:

- **Directed SoS** – The SoS is built and managed to fulfil specific purposes, typically by a controlling authority or primary stakeholder. During its long term operation, the SoS is centrally managed by that controlling authority. The constituent systems are independent, but have a clear subordinate relationship to the SoS. A typical example is the Air Traffic Control (ATC) SoS within any one country. The controlling agency plans, acquires, and manages all systems to provide ATC functions. Each system provides specific functions such as radar imaging or communications, and the entire set of systems provides safety and general air traffic control.
- **Collaborative SoS** – The SoS develops over time through collaborative work by the stakeholders of the constituent systems. As the SoS value becomes evident, the stakeholders may identify and implement a central management agency, but often that agency has no coercive power. Instead, the agency performs its management function by influence toward the stakeholders. The stakeholders and constituent systems

voluntarily collaborate for the higher-level benefits received through the existence and improvement of the SoS. A typical example is the airport, in which many stakeholders have an interest. The constituent systems at the airport may be owned by different agencies and are planned, acquired, and managed for the purposes of those agencies. Nonetheless, collaborative interfaces among the systems are implemented to enhance the overall airport functions of smooth passenger flow and security.

Figure 2 - A Modern Airport is a Collaborative SoS

- **Virtual SoS** – In this type of SoS, there is no central manager and no central authority. Individual systems interact to perform their intended purposes, and the inherent value of the SoS becomes apparent to each stakeholder. There is no "agreed" SoS purpose, but stakeholders realize the benefit of the higher-level purposes that develop. Each stakeholder modifies the constituent systems to enhance their perception of the SoS value. As a result, large-scale behaviour emerges and is desirable to the various stakeholders. A typical example is supply chain management, in which individual companies interact with each other to create a complex network of "preferred suppliers". The companies acquire and enhance software and systems to use the network. Each such installation adds to the overall SoS capability for all.

2.4 SoS Challenges

The characteristics of the SoS lead to significant challenges that are not handled well by current paradigms of systems engineering. Many of these challenges are related to "complexity", which can be defined informally as the degree to which the inter-related components of a system interact to create surprise behaviors. More formally, complexity is a mathematical concept in which a system balances between order and chaos in such a way that the behaviors become spontaneous, active, and apparently alive. There is sufficient stability for patterns to persist, but also sufficient changeability for the system behavior to respond to the environment. In this condition of complexity, there are set of acknowledged effects that present engineering challenge.

- **Nonlinear behavior** – Attempts to simplify behavior for engineering analysis often involve linearization, but complex systems often exhibit nonlinear behavior that defies analysis.
- **Positive return** – Much engineering analysis depends on entropy, the characteristic that drives systems toward a steady state. Complex systems often demonstrate periods when their behaviors move away from steady state toward new behavior sets.
- **Power laws** – Over wide ranges of parameters, complex systems can demonstrate responses that are proportional to logarithmic increases.
- **Attractors** – Complex systems have complex state maps. Frequently, there are non-intuitive regions of behavior that settle into different "attractor cycles" or "rest conditions" – but the specific attractor reached is highly dependent on the initial conditions. Sometimes, infinitesimal changes in the initial conditions can result in different attractor cycles, known as the "butterfly effect."
- **Patterns** – The SoS response is often more intuitive than analytic. Patterns appear in the behaviors, in the architectures, and in the structures that can be intuitively understood but not analyzed. (This particular challenge also provides a possible solution, in that such patterns can be catalogued and used in future SoS design, even if they are not understood.)
- **Hierarchies** – Human perception of the SoS usually used hierarchies to depict different levels of structure or behavior. The current usage of architecture frameworks relies on such hierarchies. Yet the actual SoS is "unaware" of the hierarchies and simply responds through the complicated interactions of the many components.
- **Control structures** – Without definitive analysis, control structures in the SoS often become guesswork based on intuition: "Perhaps if we try this, it will help.". Whether such a control structure actually works or

not, its usage presents a difficult challenge because of the uncertainty whether it will work again the next time.

- **Evolution** – As noted by Maier, the SoS is characterized by constant evolution. The modern airport has literally thousands of individual systems resident and interacting. Each system undergoes its own revisions for its own purposes in its own time frame. The result is that the SoS goes through constant change and evolution.
- **Trans-disciplinary concepts** – The SoS interactions transcend any one engineering discipline. Software interactions cause mechanical structures to change an electromagnetic configuration, often without any of the individual engineers being aware of the interaction. Systems engineers must be exceedingly knowledgeable to predict or even work with these challenges.
- **Emergent behaviors** – One result of all of these characteristics is the development of behaviors that exist at the SoS level. Some of those behaviors are planned, but many are a surprise to the designers.
- **Self-organization** – The complex interactions within the SoS respond to the environment to create changes in the SoS itself, such that the SoS appears to be alive, organizing itself in response to external stimuli.

3 Systems of Systems Type – DANSE Partners Perspective

This chapter aims at presenting the DANSE industrial partners SoS development examples and experience. First, a set of preliminary definitions of systems of systems aspects is provided. Then, each industrial partner has provided on or more SoS example which include some descriptive text of the SoS and the challenges that are associated with the development of these SoS.

Since some of the system's information could not be disclosed for commercial or other reasons, some projects were identified by an arbitrary acronym. Also, some information could not be disclosed.

3.1 Characterization

For the purpose of extracting initial characteristics from the SoS examples developed by the industrial partners in DANSE, the following definitions were proposed as a common basis:

3.1.1 Scope of Deployment

Scope of Deployment refers to the geographical distribution of the SoS. This includes:

- Global
- National
- Urban
- Regional
- Co-Located
- Other

3.1.2 Creation Types

Creation Types suggests descriptive options to how the SoS was realized:

- C1 Mega Systems – acquisition of very large systems, where the subsystems development is on the scale of the systems. These systems share some of the characteristics of the SoS.
- C2 Planned Evolution – the user/community has a number of systems, and desire to integrate them into SoS,
- C3 - Unplanned Evolution – user/community members performed localized/P2P, ad-hoc integration of capabilities, out of which emerged SoS
- C4 - Discovered SoS – two or more seemingly unconnected systems, whose behavior effects on another
- C5 - Significant reuse of existing sub-systems or there derivatives to create a new SoS
- C6 – Planned expansion – expand the capabilities of the SoS overtime in a planned manner by adding components or capabilities
- Other

3.1.3 System Domain

System Domain refers to the operational domain in which the system exists and provides its services or functionalities. It is not aimed at the engineering disciplines because it is understood that SoS will normally involve may engineering disciplines and technologies.

3.1.4 C4I

Command, Control, Communication, Computing and Intelligence (C4I) is an acronym originated from the military systems world. In this documents it addresses the SoS controller, if exists, with the intention of providing insight into how the SoS is operated and controlled.

3.1.5 Constituent Systems Order of Magnitude

This attribute is intended at providing a rough estimate (as opposed to an exact number) as to how many constituent systems the SoS incorporates.

3.1.6 Business Model

While Business Models suggest how the SoS is financially beneficial to the Industrial partner, it may also suggest the scope and duration vendor involvement in the life-cycle of the SoS.

3.2 IAI SoS Examples

Id	Domain	Creation Type	Scope of Deployment	C4I Architecture	CS Order of Magnitude	Business Model
A	Space	C5	Global	Centralized	5	Develop and partial operation
B	Space – Remote Sensing Satellite for Agriculture	C6	Global	Centralized	10	Develop and full operation
C	NA	C1	Regional	Cooperative	10	Develop only
D	NA	C1/C2/C5	Regional	Cooperative	10	Develop only
E	Air - Airborne Early Warning	C1/C2	Regional	Centralized	10	Develop and maintenance

Table 1 - IAI SoS Examples Attributes

3.2.1 SoS A - Satellite Communication System

The Satellite Communication System is used for relaying communication signals from one location on the earth surface to another. It employs satellite at weight between 1 to 3.5 tons, on a geo-stationary orbit which enables the satellite to remain at a constant potion with relation to the earth surface.

In addition to the satellites, the system consists of a ground control station for administration of satellites activities and operation.

Figure 3 - Communication Satallite

Satellite Command and Control Ground Stations are responsible for tracking and monitoring the satellite's equipment and performance in orbit. The telemetry received from the satellite is analyzed and processed for satellite monitoring and sub-systems status. Results are displayed on the workstations and on large screen displays.

Figure 4 - Satallite Control Room

The tracking subsystem follows the satellite through its trajectory, providing a graphic display of the satellite location. A command channel is used to introduce corrective action for any deviation or malfunction in the satellite or its sub-systems.

In-Orbit-Tests are performed to verify the compliance of the payload specifications.

The ground station is used for providing (administration of satellites activities and operation) services for other clients as well (not only those supplies by IAI).

At the users end, a dish shape antenna is used to relay the communication between the satellite and the ground based communication network.

Figure 5 - Dish Antennas

The system has a planned evolution with number of satellites increasing over time.

The development challenges identified for this system:

1. The clients usually push towards increasing the system capabilities and performance. This drives complexity and development effort and therefore drive, in turn schedule and cost up. It is therefore of interest to establish a cost effective optimum. This would require performing grand scale trade-offs for architecture, complexity and LCC. Once performed, it is also of interest to know what is the confidence level of such analysis/trade-off/optimization.
2. Lack of SoS level modeling capability for evaluating the operations and capabilities of the system.

3.2.2 SoS B - Observation satellites,

The observation satellite system differs from the "Satellite Communication System". While it includes satellites and a ground control, it does not include the user end relay stations. The ground control station is used for obtaining the remote sensing products generated by the satellites. From there the data is distributed to the customer.

This system shares the challenges of system A.

3.2.3 SoS C

The system is a complex system comprised of individual sub-systems - each one constitutes a development project on its own.

The system is deployed in its entirety, once the development phase is completed. There is planned staggered deployment over time.Most of the sub-systems are new, but are derived from similar previous generations. Each system can operate independently, or in cooperation as part of a cluster of similar systems. The CONOPS were supplied at the development initiation by the costumer.The challenges that were recorded for this SoS were:

1. Allocating System level requirements to the sub-systems (1 SRD in n SSSs)
2. Limited V&V capability at the systems level

3.2.4 SoS D

Like SoS C, this system is also a complex system comprised of individual sub-systems - each one constitutes a development project on its own.

Each system can collaborate with other similar systems to share information and resources for improved performance. Additionally, the system can collaborate with higher level systems.

The challenges that were recorded for this system were:

1. The combination of large scale system, complexity and numerous engineering disciplines/knowledge areas makes it hard to populate the SE positions with qualified professionals.
2. System level modeling and simulation
3. The magnitude and complexity of system implies very large numbers of interfaces and data flows, prone to error arising at the later stages of the development.
4. How to carry out partial system level V&V but still get confidence on the overall capability/performance?

3.2.5 SoS E

This system is an airborne early warning system. It contains an airborne platform and several ground sub-systems. The system can communicate and collaborate with other similar or higher level systems. Its command and control center controls some the sub-systems, but not all.

The sub-systems existed prior to the project start. The development consisted of the improvement of some of the sub-systems and merging all of them into one system.

No challenges were recorded for this system.

3.3 CARMEQ SoS Examples

Id	Domain	Creation Type	Scope of Deployment	C4I Architecture	CS Order of Magnitude	Business Model
A	Civil, Automotive-Industry	C2/C5/C6	International	Centralized	10	Develop and full operation

Table 2 - Carmeq Example Attributes

3.3.1 SoS A

The system consists of IT frameworks of several multinational enterprises. It is used to provide cars worldwide with updates. The time within a cars life-cycle at which the update is done ranges from the end of production line to cars already in use for several years.

Challenges:

1. The communication between the IT frameworks has to be coordinated while maintaining security levels.
2. Uptime has to be 24/7.
3. Updates are based on the configuration of a car which can differ considerably even within one car line.

4. System has to evolve along with the new developments to incorporate new car lines as well as new components to existing series. New components can be especially challenging as they increase the diversity within a product-line even more.

3.4 THALES SoS Examples

Id	Domain	Creation Type	Scope of Deployment	C4I Architecture	CS Order of Magnitude	Business Model
Military modernization programme	Defence, Military	C1/C2/C5/C4/C6	National / International	Centralized / Cooperative	From hundreds to thousands	Develop and full operation

Table 3 - Thales Example Attributes

A national or multinational military modernization program typically uses military-run experimentation for acquisition of new armaments, and tests new military behaviors such as effects-based operations, network-centric warfare, and irregular warfare and civilian-military concepts.

The combination of the compositions of the forces and the contexts of their deployment are impossible to describe in an exhaustive way, because the missions are forever changing and the preparatory work, which precedes the deployments on the theatre of operations, has to take into account, not only the targeted final effect, but also the available men and the projectable or pre-positioned means. Thus, the user need is impossible to specify because it is only very partially known when the call for tender is published. It results from these constraints (the other parameters can naturally act), that it is very difficult for the customer to specify without ambiguity the required technical object “system of systems”.

The main challenges are of three different natures:

1. Operational and technical stakes, linked to the complexity and to the variety of the commitments and the threats;
2. Organic stakes, linked to the transformation of the organization and its supporting means; this includes the establishment of a trusted relationship with the operational teams so as to be able to set up the conditions of the success of the integration (human factors), from the first ideas until the deployment and the training;
3. Financial stakes, linked to the requirement of coherence of all the investments, in particular with respect to the capacity needs.

To address these challenges, the implementation of an iterative process of development is necessary. This iterative process will drive the renovations, the developments and the projects in order to converge from the existing/legacy systems to the targeted architecture. The process of iterative development will have to allow for the integration of new features and new platforms, in a mastered way and without having to re-qualify the whole system of systems.

To address the technical facet of these challenges, the following high-level engineering goals need to be reached:

- Definition of the engineering roadmap;
- Analysis of the needs: analysis of the operational needs and definition of the capacity roadmap;
- System of systems architecture: definition of the overall architecture, definition of the increments, definition of the applicable norms/rules[2], data model, service model, interfaces, tests baseline;
- System architecture: definition of the critical/common component systems, and for each, definition of the applicable norms and rules, data model, service model, interfaces, tests baseline.

Typical approaches include:

[2] Including the customer’s choice in terms of architecture framework, e.g. DODAF, MODAF, NAF, etc.

- Definition, between the Tasking Authority and the Lead SoS Architect / Integrator, of the representative use cases, to make possible the contractualization, the study, the implementation, and then the qualification of the system of systems; typically, operational scenarios can be used as basis to the commitment of the industry when the associated performances are quantified; other intrinsic characteristics to the system of systems (e.g. adaptability, modularity, evolutivity, etc.) will be of specific interest, because they will render possible its adaptation to the various deployments to face the regular emergence of new operational needs; a side-effect of this approach is that the technical solutions supporting the architecture are not frozen in the contract, but can evolve with the operational needs, upon request, whilst maintaining the overall consistency;
- Digital and hybrid simulation to characterize the operational synergies and the best way of operating them, as well as to validate the doctrine, improve the training and the support.

Another significant challenge for a typical armament operation is in the necessity to demonstrate and quickly validate the obtained capacity gains on a set-up representative of a real deployment.

3.5 EADS SoS Examples

Id	Domain	Creation Type	Scope of Deployment	C4I Architecture	CS Order of Magnitude	Business Model
C2	Defense, Military, Civil	C1/C2/C5/C6	Global/Regional/ International	Centralized	10	Develop and full operation

Table 4 - EADS Example Attributes

A Command and Control system (C2 system) is a combination of multiple complex systems which include human systems, technical systems, and services systems. The constituent systems collaborate in order to achieve a predefined mission or purposes.

Such a system of systems can be applied to different fields of applications. In defense for example, the process of identifying and predicting threats depends on different kinds of surveillance and satellites systems, which provide information about the targeted area. The information will be processed by another type of systems that includes human interaction to give decisions, while the actions and orders are implemented by other systems.

Another application in the civil field is the Emergency Response System. The police command and control together with the fire brigade and medical systems, try to solve and mitigate the consequences of emergency cases. The Emergency Command and Control Center organizes the work between different systems and distributes the missions and orders. The communication between these systems is achieved by communication services systems that provide the Cell phone, radio, phone, and internet and network services.

Challenges outstanding in the above SoS:

1. Constituent systems complexity and interaction:
2. The interaction between human systems and technical systems
3. Communication channels and information exchanges between constituent systems and the integration of new communication technologies: the new communication technologies will provide new capabilities that will enhance the decision making process and decrease the response time. However the effect and the integration of the new technology can lead to positive or negative emergent behavior, i.e. unforeseen benefits/ capabilities or threats.
4. The introduction of new constituent system to an existing C2 system: introducing new constituent systems may require new communication channels and new capabilities that will be added to the system.
5. The synergetic merge of constituent systems capabilities: the strategic planning that considers the benefits of all constituent system capabilities and join these capabilities in a synergetic way.
6. SoS evolution in the context of constituent systems evolution: each of the constituent systems manages its own evolution strategies that meet its goals and keep the system updated with the new

technologies to enhance its efficiency. This kind of evolution must be planned and coordinated between all the constituent systems considering the C2 SoS structure and rules.

7. Modeling and simulation: many issues arise in the SoS modeling field like the level of abstraction of constituent systems, human behavior modeling, system interaction, system boundaries, etc.

In addition to the above, a survey amongst EDAS project managers yielded the following challenges:

- Usually there is limited project budget available to support thorough modeling of SoS Architectures, resulting in incomplete or "just enough" SoS Modeling,
- SoS requirements and consequently design inconsistencies cannot be efficiently reworked,
- Difficulties to understand SoS-ilities and their effect on the architecting during the requirements engineering phase,
- Projects are lacking a common methodology and guidelines for SoS modeling (i.e. notations, semantics, languages, etc.),
- Method and tools are lacking to investigate high level dynamic behavior of component systems' interaction,
- Lack of human behavior considerations in SoS Design and Modeling,
- Lack of tooling framework that allows a seamless workflow of SoS Design.

Discussions with NATO also provided a set of challenges regarding interoperability in SoS that are relevant from the customer point of view:

- How to ensure interoperability while integrating new CS in SoS environments (possibly provided by different programs),
- How to include interoperability parameter and measures as integral part of the SoS design such that a sensible impact on enabling/averting synergy of the participant systems may be maximized.

4 Systems of Systems Development and Deployment Challenges

4.1 Discussion of SoS examples

4.1.1 SoS Attributes

When reviewing the SoS examples material provided by the industrial partners and looking at SoS attributes (as listed in Para. 3.1) that were filled, no clear pattern may be identified. This may be explained by one of the following:

1. The amount of data is insufficient for a pattern to be identified
2. The SoS characterization do not fit within this preliminary set of suggested attributes

4.1.2 SoS Challenges

While not being able to find a characterization defined by the set of suggested attributes, when looking at the challenges recorded, there is a certain group of challenges that were cited by most of the industrial partners.

The SoS examples challenges may be summarized and grouped as follows:

IAI:

1. Modeling and simulation at the SoS level: appropriate methods and tools to enable cost effective modeling and simulation at the SoS level to gain sufficient understanding of the SoS behavior and attributes.
2. V&V - Obtaining enough confidence at the SoS level when performing partial or limited V&V.
3. Management of very large number of interfaces throughout the development and integration phases.

Carmeq:

1. Maintain IT availability, security, and configuration control across multiple networks.
2. Adapt to changes in constituent systems.

Thales:

1. User needs that are partially known at the beginning and are continuously changing, combined with complexity suggest increased development effort.
2. Organizational support for instrumental cooperation with the operational teams throughout the program.
3. Meeting the financial objectives while still addressing the capacity needs.

EADS:

1. Identifying unforeseen benefits/ capabilities or threats (positive or negative emergent behavior) that stems from complexity and interaction between CS, and between users and/or operators with CS
2. Means for strategic planning that considers the benefits of all constituent system capabilities and join these capabilities in a synergetic way, considering the foreseen evolution of the CS.
3. Obtaining modeling and simulation means that address key aspects of the SoS while providing meaningful results with high level of certainty.

The above grouping of challenges that were cited by the industrial partners provides the project with a firm basis for guiding the research products expected in DANSE.

The need for **modelling and simulation means** at the SoS level, which clearly stands out as a common challenge in the examples of the industrial partners, is a reflection of some significant aspects the SoS and its development. This need is driven from the large scale and complexity, the difficulty in predicting its (expected or unexpected) **emerging behavior** because of the interaction between the constituent systems.

Although it has been mentioned only by EADS, the difficulty to understand SoS-ilities and their effect on the architecting of the solution is worth noting and must be taken into account in the WP5.

4.1.3 Correlation of Challenges

4.1.3.1 INCOSE SE Handbook

In the systems engineering overview chapter of the INCOSE "SYSTEMS ENGINEERING HANDBOOK" (INCOSE-TP-2003-002003.1, Aug 2007), the following challenges are associated with systems of systems:

1. **System elements operate independently**. Each system in a system of systems is likely to be operational in its own right.
2. **System elements have different life cycles**. SoS involves more than one system element. Some of the system elements are possibly in their development life cycle while others are already deployed as operational. In extreme cases, older systems elements in SoS might be scheduled for disposal before newer system elements are deployed.
3. **The initial requirements are likely to be ambiguous**. The requirements for a system of systems can be very explicit for deployed system elements. But for system elements that are still in the design stage, the requirements are usually no more explicit than the system element requirements. Requirements for SoS mature as the system elements mature.
4. **Complexity is a major issue**. As system elements are added, the complexity of system interaction grows in a non-linear fashion. Furthermore, conflicting or missing interface standards can make it hard to define data exchanges across system element interfaces.
5. **Management can overshadow engineering**. Since each system element has its own product/project office, the coordination of requirements, budget constraints, schedules, interfaces, and technology upgrades further complicate the development of SoS.
6. **Fuzzy boundaries cause confusion**. Unless someone defines and controls the scope of a SoS and manages the boundaries of system elements, no one controls the definition of the external interfaces.
7. **SoS engineering is never finished**. Even after all system elements of a SoS are deployed, product/project management must continue to account for changes in the various system element life cycles, such as new technologies that impact one or more system elements, and normal system replacement due to pre-planned product improvement.

The table below assigns (subjective) correlation measure between the modeling and simulation challenge, and its associated emergent behavior prediction difficulty, identified in the DANSE industrial partners SoS examples and the challenge identified by INCOSE.

Challenge	Correlation*
System elements operate independently	H
System elements have different life cycles	L
The initial requirements are likely to be ambiguous	M
Complexity is a major issue	H
Management can overshadow engineering	L
Fuzzy boundaries cause confusion	M
SoS engineering is never finished	M

H = High, M = Mid, L = Low

Table 5 - SE Handbook Challenges Correlation

It can be seen that the need for better modeling and simulation means underlies the majority of the SoS challenges listed in the INCOSE SE handbook.

4.1.3.2 INCOSE SoS Working Group

The INCOSE SoS Working Group (SoSWG) had recently made public the initial results of a "Pain Points" survey, aimed at identifying candidate areas for SoSWG initiatives. The survey asked respondents to identify and describe their priority SoS areas of concern.

There were 38 respondents to the survey and these respondents report 65 ‘pain points’. The vast majority of the respondents were from the US (86%). The UK (8%) and Australia (6%) also made significant contributions. Almost all report extensive (60%) or some (37%) experience working with SoS. Almost all are working in the defense sector (94%).

The results of the survey were reviewed and sorted into major affinity groups, with primary division into two broad categories: management and technical.

The affinity groups within the technical category (40 out of 65 pain points) are listed below (values indicated percentage numbers of pain points out of 65):

- **Emergence** (6%, 4/65) - In this area, respondents focused on the recognized SoS challenges of expected behavior from an SoS based on the resultant combination of systems.
- **SoS Interdependencies** (11%, 7/65) - A related set of responses address the complexity of understanding the complex relationships among systems in an SoS highlighting both the difficulties of understanding these interrelationships and the lack of methods for their analysis.
- **SE Processes** (25%, 16/65) - About a quarter of the responses addressed issues with applying today’s SE processes to SoS, starting with requirements through configuration management and testing. Several areas here were very defense centric such as comments concerning security and information assurance. Others (Configuration Management and Cost Estimation) were noted but by only by single respondents. However, there were two areas with multiple respondents on areas of broad SoS applicability.
- **Capabilities and Requirements** (9%, 6/65) - In this area, a number of respondents reflected that the tradition approach to requirements in SE of systems poses issues for SoS. In an SoS context, many people prefer to focus on capabilities and less on requirements.
- **Testing, Validation, and Learning** (8%, 5/65) - This is another recognized area of SoS challenges reported by multiple respondents in the survey. Comments reflect the fact that most defense SoS cannot be tested thoroughly prior to fielding leading to approaches like incremental validation, reflecting a perspective that looks at significant learning going on over the life of an SoS.

- **SoS Thinking Principles -** This final area included issues (8%, 5/65) related to factors which respondents indicated were either missing (negative) or needed (positive) for successful SoS.

As in paragraph 4.1.3.1, the table below assigns (subjective) correlation measure between the modelling and simulation challenge and the pain point identified by SoSWG.

Pain Point group	Correlation*
Emergence	H
SoS Interdependencies	H
SE Processes	L
Capabilities and Requirements	M
Testing, Validation, and Learning	H
SoS Thinking Principles	L

H = High, M = Mid, L = Low

Table 6 - SE SoSWG Pain Points Correlation

As with the INCOSE SE handbook SoS challenges, the majority of the pain point groups are highly correlated with the need for better modeling and simulation.

4.1.3.3 DANSE Project Requirements

DANSE technical requirements are documented in deliverable 3.1. The project technical requirements (a total of 76) relate to DANSE Technical scope, User Needs, Examples, and Technical solutions as sketched in the diagram below.

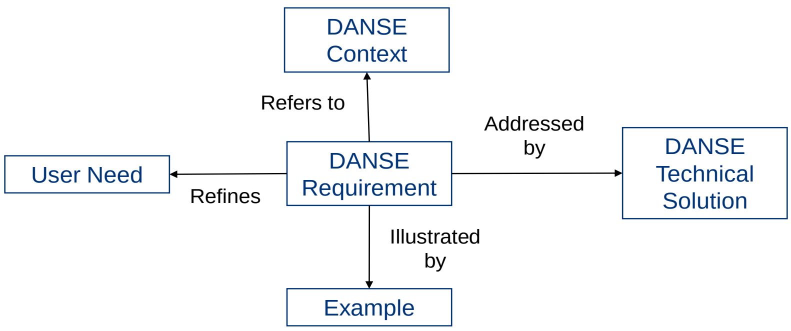

Figure 6 - DANSE D3.1 Requirements Relationship

DANSE research plans to address five main technical challenges, as described below:

Methodology - The size, complexity, and dynamicity of SoS result in serious difficulty corralling the component systems to perform the desired services while creating frequent incidents of unexpected emergent behaviour. Today's top-down and once-through systems engineering processes become inaccurate and unreliable under these circumstances.

Modelling - There is no formal semantics for SoS modelling and design languages. Consequently, descriptive models are ambiguous and there is no disciplined technical basis for predictive models that support architectural trade studies and operations analysis.

Simulation and analysis - 1a) Component-based simulation of SoS creates large simulation models that are often subject to combinatorial explosions. 1b) Simulation engineers at the SoS level suffer from a lack of purview over the structure of component systems. 2) Comprehensive virtual testing of an SoS is often prohibitively expensive.

Architecture - The limits of what can be known about SoS that is an opportunistic aggregation of legacy systems imply that frequent instances of unexpected emergent behaviour are inevitable, making architectural processes chaotic and ineffective.

Tool Net - SoS are typically geographically distributed and involve loosely coupled organizations. Today, vast collaborations are characteristically inefficient and error-prone. It is difficult to get SoS design and operation teams to coordinate their actions and for SoS managers to have a global view of SoS behavior, thus creating a lack of situation awareness that can lead to significant problems.

Based on these definitions, one of the five technical challenges was assigned to each requirement (though a requirement may relate to more than one technical challenge). The number of requirements that are associated with the modeling or simulation technical challenges accounts for about third of all the requirements, as presented in the following chart:

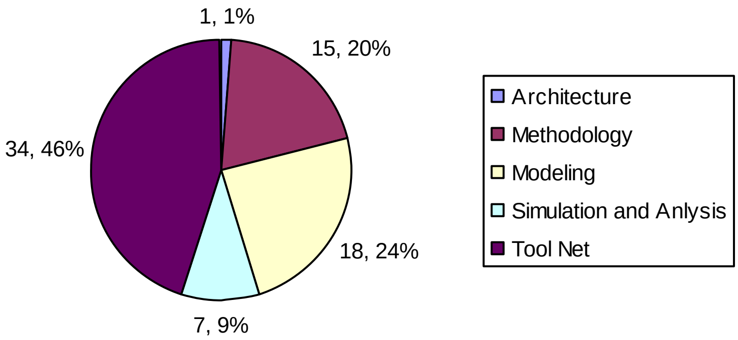

Figure 7 - DANSE D3.1 Requirements to Technical Challenges Distribution

But, while considering that fact that a requirement may relate to more than one challenge, and reviewing requirements textual description, the number of requirements that are associated with either modeling or simulation account for half of the requirements.

4.1.3.4 Complexity

Paragraph 2.4 addresses the role of complexity in presenting significant challenges that are not handled well by current paradigms of systems engineering.

These challenges include Nonlinear behavior, Positive return, Power laws, Attractors, Patterns, Hierarchies, Control structures, Evolution, Trans-disciplinary concepts, Emergent behaviors and Self-organization (see 2.4 or more details).

It can be seen the majority of these challenges highly correlates with modeling and simulation challenges and may greatly benefit from improved such capabilities.

4.2 Conclusions

The work on this chapter began by asking the industrial partners to provide descriptive information from SoS projects within the respective organizations. For this purpose, a set of suggested preliminary SoS attributes have been prepared (see 3.1). Additionally, the industrial partners were asked to provide information about the challenges that they have encountered during the development of those SoS.

The provided information about the SoS attributes did not amount into a meaningful observation. While disappointing, this concurs with the many definitions and characterization attempts made before DANSE. However, the need for (better) modeling and simulation means at the SoS level clearly stands out as a common challenge in the examples. This also reflects in the correlation to the challenges addressed in other works, as presented in 4.1.3.

So while it was not possible to derive a direct meaningful insight about SoS themselves, we were able to identify a significant void in the SoS prime contractor toolbox, one that may certainly affect other SoS stakeholders.

5 Summary

This document has reviewed existing types of SoS developed and/or operated by the DANSE industrial users and characterized them according to different criteria, as a basis for directing the new DANSE SoS Engineering paradigms to the user needs.

As the literature in SoS is growing rapidly and the precise definitions are subject to change as the body of knowledge becomes more established, the document also presents an overview of current definitions from the available literature on SoS Engineering.

Anchoring the challenges associated with SoS development and evolution through the outlining of a set of industrially relevant SoS examples has served to provide tangible context for the DANSE project as a whole. In section 3 it will be noted that across the industrial domain there are many common perspectives to the challenges that lie ahead. IT is very clear that iterative development will be essential in representing a SoS and performing first high-level modeling, simulation and analysis down to Constituent Systems detailed specific simulation in order to address the global optimization and its possible conflict with local optimizations. Even if each component system were optimized there is no guarantee that the overall SoS will operate in an efficient manner.

To obtain practical results from the above characteristics, section 4 highlighted industry partner specific identified challenges for a SoS. Actually, all these seem to be of concern to all the industry partners and further raise the importance of the requirement for an integrated tool chain that supports modeling, simulation and analysis in both breadth and depth. A detailed mapping of the challenges to the requirements and user needs document is presented to serve as a reference for directing the research work in DANSE towards practical industrial needs.

In summary the need for modeling and simulation means at the SoS level, clearly stands out as a common challenge in the examples of the industrial partners, and is a reflection of some significant aspects the SoS and its development: their large scale and complexity, the difficulty in predicting its (expected or unexpected) emerging behavior because of the interaction between the constituent systems. Although it has been mentioned only by one industrial user in DANSE, it is worth noting the difficulty to understand SoS-ilities and their effect on the architecting of the solution and they must be taken into account in the work on SoS architecture patterns to be performed in the project.

The application of the DANSE project research and innovations to tackle real industrial problems will help draw out the critical aspects that a future SoS enterprise will need to address both from a technical, people and organization perspective both at the DANSE industrial partners future SoS business, as well as for the broader European SoS community.

The significance of the DANSE project is huge, not only in its ability to raise awareness, stimulate debate and discussion, but also in the development and application of a comprehensive suite of SoS models and tools that operate within a robust methodological framework and an effective integrated tool environment. This document as anchored these directions on practical examples and cross correlation to the existing literature. The stakes are very high because our dependence on SoS is growing and any malfunctions on their part could have devastating consequences for society.

6 Abreviations and Definitions

C4I	Command, Control, Communication, Computing and Intelligence
CS	Constituent System
DANSE	Designing for adaptability and evolution in system of systems engineering
DoD	Department of Defense
DODAF	Department of Defense Architecture Framework
INCOSE	International Council of Systems Engineering
IT	Information Technology
LCC	Life Cycle Cost
MODAF	Ministry of Defense Architecture Framework
NAF	NATO Architecture Framework
NATO	North Atlantic Treaty Organization
SE	Systems Engineer/ing
SoS	System of Systems
SoSWG	System of Systems Working Group
SRD	Stakeholders Requirements Document
SSS	System/Sub-System Specification
TOGAF	The Open Group Architecture Framework
V&V	Verification and Validation

7 References

DANSE, 2012	Gérard Cristau, Stéphane Paul, Henry Broodney, Uri Shani, Tim Lochow, Leonardo, Mangeruca, Alberto Ferrari, Roy Kalawsky, Christoph Etzien, Thomas Peikenkamp, Juergen Mueller, Eric Honour, Nir Ben-Dov, Zvi Lando, "Project Requirements", DANSE Deliverable D3.1
Manthorpe, 1996	WILLIAM H. J. MANTHORPE, JR, The Emerging Joint System of Systems: A Systems Engineering Challenge and Opportunity for APL, Johns Hopkins APL Technical Digest, Vol. 17, No. 3 (1996), pp. 305–310
Jamshidi 2005	Mo Jamshidi, "System-of-Systems Engineering — A Definition," IEEE SMC 2005
Schonenborg, ESA 2010	R. A. C. Schonenborg, T. Bieler, A. Matthyssen, M. Fijneman, System of Systems architecture in ESA's Concurrent Design Facility, SECESA 2010
Maier 1998	Maier, M.W., "Architecting Principles for System of Systems," Systems Engineering, Vol. 1, No. 4, 1998, pp. 267-284
INCOSE, 2007	INCOSE, "SYSTEMS ENGINEERING HANDBOOK", INCOSE-TP-2003-002003.1, Aug 2007
INCOSE 2012	INCOSE Systems of Systems (SoS) Pain Point Survey