Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Toward a Science of Resilience, Supportability 4.0 and Agility

Peter Bernus, Ovidiu Noran, Ted Goranson

IIIS Centre for Enterprise Architecture Research and Management, Griffith University, Australia, [P.Bernus, O.Noran, T.Goranson]@griffith.edu.au

Abstract: Large and complex systems have a long-expected life and evolve slower than small systems. As such, they may live through several technological, social, economic and ecological changes in their environment. A fundamental challenge discussed in this paper is how to (re)design and change large-scale systems so that they remain maintainable and evolvable e.g. for the expected duration of their lives. This must also be achieved in view of their legacy and carried out in an affordable, risk-mitigated and timely manner. After defining some important features of large-scale systems and reviewing the state of the art in managing systems evolution, the paper characterizes the problems, solution scope and opportunities in the area and defines basic principles, theory, associated life cycle architecture and methodology approach for long-term systems supportability.

Keywords: Enterprise Integration, Resilience, Supportability, Agility, Systems Engineering, Complex Systems.

1. INTRODUCTION

1.1 Characteristics of the lives of large-scale systems

Large and complex systems typically live long. The expected lifetime is often 30~60 years, and some systems have even longer (indefinite indeterminate) life spans.

Typical examples are various infrastructure systems (such as energy, water, road networks, communication systems, space systems), large production facilities (such as manufacturing), complete supply chains, transport systems (civilian and military aircraft, ships, railways, road transport) etc.

The speed of evolution of these systems *on the system level* is understandably much slower than the evolution of various constituent smaller systems (such as sub-systems and components, or contributing systems in a system of systems).

Therefore, during their lifetime these systems live through many technological, social, economic and ecological changes that occur in their environment. These may be due to new demands, such as having to satisfy *new functional and non-functional requirements*, or due to *changes in the supporting systems* these systems depend on (ISO15288, 2015).

Such a large system is therefore best considered as a *living system* that evolves organically. Notably, for survival, it is imperative to have the ability to change in an agile way so as to adapt to the above types of changes, while at the same time preserving the system's fundamental identity – otherwise the system is forced to be decommissioned and will cease to exist.

A fundamental challenge discussed in this paper is how to architect (or, more commonly, how to (re)design and change) a large-scale system so it remains maintainable and evolvable e.g. for the next 40+ years. This must be achieved in spite of the heritage of the past (like embedded code in a typical banking system, airliner, or in the control systems of power stations, railway systems, etc.), and carry out this transformation in an affordable, low risk, and timely manner.

Organisations that operate complex systems of this nature require that these systems (after having been deployed) have the following characteristics:

- remain *Supportable* for a long period of time,
- endure external or internal changes in a R*esilient* way (i.e., stay operational in spite of various adverse or unexpected influences),
- adapt in an A*gile* way to already occurring or anticipated changes in circumstances, i.e., do this fast enough to keep satisfying the requirements (irrespective of whether the system's transformation was planned or forced).

The above 'triumvirate' inspired the title of this paper, but we note that i) there are several other important system 'ilities' (Boehm, 2014) that we do not discuss here, despite their importance, and furthermore ii) out of these three, our focus in this paper will be *supportability*.

The timeliness of the problem statement is stressed by the fact that the industrial world is undergoing a major change on multiple fronts, sometimes referred to as the fourth industrial revolution ('Industry 4.0', 'Smart Manufacturing', etc.).

There exists a list of strong technology-based opportunities that promises new ways of production and through-life support, potentially leading to more dynamic and at the same time more predictable, dependable and optimised supply chains, paired with better informed and faster decision making on multiple time horizons (from strategic down to real time).

In a sense, these opportunities promise a qualitative change to a market player's effectiveness and efficiency through improving the player's 'OODA loop' (Osinga, 2006). This has been shown to be the most important critical success factor of winning in a competitive environment. (Note that this is true even though on closer investigation OODA is not really a simple loop, but a complex process consisting of a network of interconnected parallel activities.)

1.2 State of the art in Managing Systems Evolution

Resilience

Resilience is defined by the UN Office for Disaster Risk Reduction (UNDRR, 2017) as "*the ability of a system, community or society exposed to hazards to resist, absorb, accommodate, adapt to, transform and recover from the effects of a hazard in a timely and efficient manner...*".

Resilience is often associated with the system's ability to gracefully respond to sudden adverse events, but the concept also includes the ability to deal with the often more dangerous 'slowly creeping' detrimental influences, whereupon a delay in recognising the danger denies the affected systems the ability to recover.

As time progresses, large and complex systems – the subject of study in this paper – may undergo changes due to multiple reasons. i) Adjustments may be necessary due to changes in the environment in which the system is embedded (because of new demands or opportunities, new threats, changes in the technical, social, ecological, economic, political and organisational relationships to the external world, including the relationship to supporting systems (ISO15288, 2015); or ii) Adjustments may be required due to internal factors, such as planned evolution of the system's capabilities to meet the future environment's demands, or due to other internal effects.

For the above reason, the static picture that describes a system as 'being resilient' is insufficient – even though being resilient at a point in time is a useful criterion for a baseline assessment, the concept must be enriched by a meta-concept that describes how a system preserves, increases (or decreases) its resilience over time.

Supportability

Supportability is defined as *"... the inherent characteristics of the system and the enabling system elements that allow effective and efficient sustainment (including maintenance and other support functions) throughout the system's life cycle*" (DAU, 2010; p205).

According to (GAO, 2011; p182) support of complex military systems / platforms often account for 70% of the total life cycle cost. Unfortunately, the margin of error in estimating this cost is high, but it is usually underestimated.

Nevertheless, early considerations of supportability in the system's life are known to have the biggest impact in terms of a number of desired characteristics, such as readiness, availability and cost (ISO16091, 2018; IEC60812, 2018).

The literature on supportability is extensive, covering various techniques (e.g., condition based- and predictive maintenance), pertinent design principles, programme & project management processes, supply chain management practices and organisational models.

Each of these has a role in achieving optimum supportability, many of the problems are well understood, and (standard) processes to address them are in place in the framework of Integrated Logistic Support (ILS) (Jones, 2006). However, in case of large-scale long-lived systems there exist special requirements, namely that the supporting system (that is a socio-technical Systems of Systems itself) *must also be supportable*, and this aspect is not well developed in ILS.

Therefore, the same observation as for resilience can be made: for the ILS of long-lived systems the current picture seems still static in the sense that it assumes too much predictability of future supportability demands. The reason of thus gap is that these large systems are expected to survive several technology revolutions. As a result, planning on the basis of current technology has limits, and there is a need for a solution to break out of this limitation, and the present article attempts to propose approaches to tackle the problem.

Agility

The term is used throughout the industry in two somewhat different ways: a) the ability of the system to *modify itself* to adapt to current or future needs, and b) the ability of the system to perform adjustments *fast* (e.g., in new product development, bespoke development, manufacturing or service delivery) and maintain a constant feedback loop with stakeholders so as the solution converges to what is desired. The two are naturally related because 'b' assumes that the system (enterprise) is able to adaptively change so as to be able to perform as needed.

According to Dove (1995, 2005), there are two fundamental ways to respond in an agile way. The *proactive* response category includes the ability of the system to develop in order to meet a new need, the continual improvement or upgrade to increasing response-performance needs, the migration to a better approach as required, or modification of a system by inserting subsystems with new capabilities.

Within the *reactive* response approach one can include the correction of a malfunctioning subsystem or unintended consequence, the response to input or response-need variation, a response to increased or decreased capacity needs, or the reconfiguration of subsystem relationships.

Agility metrics include time cost, quality and scope. Quality refers to robustness (at least sufficient rather than as an interim convenience) and predictability (i.e. the response being also as specified). Scope emphasizes the ability of agile systems to accommodate response requirements beyond a strictly predefined set of options.

1.3 Problems, Solution Scope and Opportunities

Systems engineering methodologies stress that during the early stages of a system's life the through-life supportability needs must be taken into account, and suggest a number of techniques for addressing this requirement.

These techniques include the statement of supportability characteristics in form of design principles as well as the formulation of relevant functional and non-functional requirements.

Traditionally these requirements were stated in qualitative terms, but lately (i.e., the recent twenty years) the systems engineering literature started to stress the importance of stating these to be quantitatively measurable. Note that 20 years is 'recent', because we are concerned with systems or their components that have been around considerably longer.

As an example, the US Government Accountability Office analysed the sustainment challenges of more than 2,800 selected fixed wing military aircraft (GAO, 2018) and concluded that major causes of failing to meet availability goals were all due to supportability issues, such as:

- Delays in acquiring replacement aircraft
- Unexpected replacement of parts and repairs
- Delays in depot maintenance
- Shortage of depot maintainer personnel
- Parts obsolescence
- Diminishing manufacturing source

The report concluded that depending on platform-specific needs there can be different approaches to increase availability trough improvements to the support system.

Unfortunately, the report does not offer a generalisation of this finding, but we believe there should be, and that the outcome could then be used in planning support for any long-lived system. Also, given that some of these systems are more than 50 years old and are expected to remain operational for the foreseeable future (e.g. the B52), it is unrealistic to expect that the support system can be improved by a single one-shot effort.

To summarise, both the *platforms* of concern and the *support system* are living systems, and therefore any solution must be based on an underlying *enduring architecture* that allows both systems to co-evolve with the environment (Kandjani et al, 2013), but without losing identity.

In fact, apart from these two obvious systems of interest, the chain of interrelated systems to be considered is even longer, so the *scope of the problem* goes beyond the above two. Other systems of which the above properties play an important role in achieving optimal supportability include:

- the acquisition system that *defines* capability needs and *acquires* platforms,
- main contractors and their supply chains that *design* and *deliver* the capabilities (NB. the support system is often considered part of this chain, but as discussed in Section 3 this is not necessarily the case),
- entities that *operate* the capability (themselves often a complex system of systems), but also contribute to *acquisition decisions*,
- government entities that make *political* decisions, and *portfolio* owners,
- institutions and industries that *feed* the above systems, such as science, R&D, technology, and those that provide education & training (workforce),
- various *regulatory* entities.

The objective is to develop methodologies, design principles and techniques that can be used by the parties involved to guarantee and to maintain the desired systemic properties (the 'ilities') of long-lived systems of systems throughout their lives, and can do so using a long-term perspective.

Some challenges that a comprehensive theory of the problem space should address include the following:

- during the life of the systems of interest our technologies evolve, therefore fast technology insertion is often necessary for the system to be able to co-evolve with the requirements,
- human skill sets change, and there may be little or no expertise left for maintaining or repairing a legacy system (e.g., languages used for programming change over decades);
- some supply chain participants may be out of business or have no capability left to contribute a service;

There exist new technological, organisational and architectural change opportunities that can be explored. For example, the recent decade of developments in massive-scale sensor- and data processing technologies are an enabling factor for supply chain-wide information sharing never experienced before. Figure 1 lists technologies, movements and approaches likely to be considered for insertion into our systems of interest.

However, this needs to be combined with complementary organisational and architectural developments (otherwise one cannot expect to achieve long lasting capability effects).Importantly, one of the architectural design requirements to adhere to is that large-scale systems must be able to be transformed while they are operating.

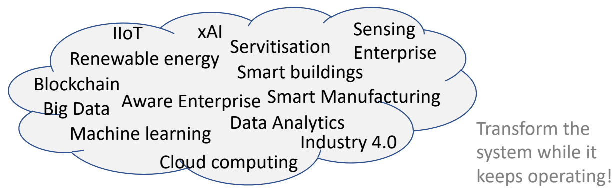

Fig. 1. Technologies waiting to be injected

An example for the enablement of long-lasting underlying systems architecture is NATO's technology agnostic Deployable Communications and Information System (DCIS) Cube Architecture (NATO, 2018). DCIS's architectural design achieves this by deliberately decoupling the functional components of the system into layers that can independently evolve in the future.

2. DEVELOPING A SCIENCE OF SUPPORTABILITY

The present research is based on conceptual-analytical study, and is aimed at inspiring further work in its space of enquiry rather than attempting to present a finished result that is to be ready to be used out-of-the-box.

The basis of this study includes a multitude of sources, from several disciplines, due to the fact that the problem requires interdisciplinary and even transdisciplinary contributions. After all the systems in question are embedded in a larger socio-technical context, where the success of their continual transformation depends on *interacting* organisational, social, economic, cultural and political factors, in addition to purely engineering considerations.

For example, the success of applying large-scale deployment of sensors, as well as data analytics and artificial intelligence technologies for improved supply chain dynamics hinge on many features, such as trust among supply chain partners, their legal and organisational preparedness, matching capability and maturity levels, willingness to implement (or if not available then to develop) standards, as well as the ability to influence regulatory reforms.

Cultural challenges include the need to improve capability planning and acquisition decision making that is currently still dominated by quantitative investment budget considerations (with the propensity to shift the burden to the future support system), and a better integration of actual operational information into engineering design decisions.

3. SUPPORTABILITY 4.0: AN INITIAL PROPOSAL

The target of this proposal is identified as a *science of supportability* that must include a theory (axioms that lend the theory its pre*dictive* and *explanatory* power, and which can be used to construct reliable and feasible methodologies, techniques and processes, for example), and a life cycle architecture (which is the subdivision of the conceptual elements of the problem space in a way that is *complete in scope*, setting the scene for a terminology the theory can use).

A note on completeness of scope: the systems in question are inherently open (in terms of no explicit and at the same time complete enumeration of all future elements, components, interactions, aspects and perspectives is possible or even desired). As such, the best one can do is to define open subdivisions of scope, such as for example: the scope must include 'everything that is automated and everything that is not', or 'everything that is done by hardware and everything that is not', 'every function the system needs to fulfil its mission, and everything else the system performs for itself (management, command, and control)', etc.

The authors shall use the term 'solution' below in a generic sense, to mean 'the architectural design of a system' – i.e., of some entity of interest that embodies the system. E.g. 'solution' may mean the architectural design of a supply chain providing support services to a class of products, it may mean the architectural design of a product line of vehicle platforms, or refer to the architectural design of a type of communication system, as the case may be (because the supportability of any and all of these can be a matter of interest to stakeholders).

3.1 Supportability 4.0 Principles

Below, the authors define some principles that a theory and associated life cycle architecture for long-term systems supportability must satisfy:

- The solution must be *technology agnostic*, so as to remain updatable in light of future developments;
- The architecture needs to encompass in its scope the life cycles of all significant related systems as discussed in Section 1;
- The solution must enable complexity management of the involved system of systems to ensure *viability* and *controllability* (Bernus et al, 2016);
- Any architectural solution needs to account for systems that are *agile*, dynamic and adaptable, balancing the enduring immutable architecture features with those that are constantly updated and configured during operation;
- Solutions must be able to be evaluated both qualitatively and quantitatively with regard to the *resilience* of the Systems of Systems in the scope;
- Solutions must be able to be evaluated against *economic* effects present in a system's life (on micro, meso- and macro levels);
- Solutions must account for *human* and *social aspects* of evolving systems and be available against relevant values, norms, and other stakeholder expectations (Romero et. al, 2016);
- A Solution needs to consider system 'ilities' both as designed features of a system architecture and as emergent properties;

The claim here is that the above list is a relevant sample, but we expect that it will have to be further developed.

Potentially all entities of interest (see Section 1.3) need to be considered by the theory; therefore, we shall represent the theory in a generalised form, which then can be specialised by considering existing and predictable pressure points whenever the supportability of a particular large-scale system is being considered.

Figure 2 represents so-called *life-cycle relationships,* where each arrow depicts the fact that one entity performs some life cycle process of another entity. This includes the situation describing that i) entity A is changing entity B (or itself) [solid arrows], and ii) entity A provides support to B or otherwise interacts with B [dashed arrows].

A complete model would then have to represent the responsibilities for *each* of the life cycle processes of *all* of these entities; if no responsibility is clarified, then there is no control exercised and there is no guarantee for enduring supportability.

The way to read the model in Fig. 2 is best explained through examples. E.g., the two solid arrows starting at the Operation of the virtual MRO (vMRO) show that the vMRO performs (re)design / (re)build / (re) configure the Platform, as well as performs some decommissioning activities (such as decommissioning part of the platform due to replacement of some component). The dashed arrow starting from the Operation of the Platform shows that the Platform operationally supports (is utilised in) the Mission Fulfilment System.

Note that Fig. 2 abstracts from time, therefore a separate model will be necessary to represent and to analyse *when* and how often the instances of these life cycle processes need to be performed (this is called a 'life history model' – *c.f.* ISO15704 (2019)).

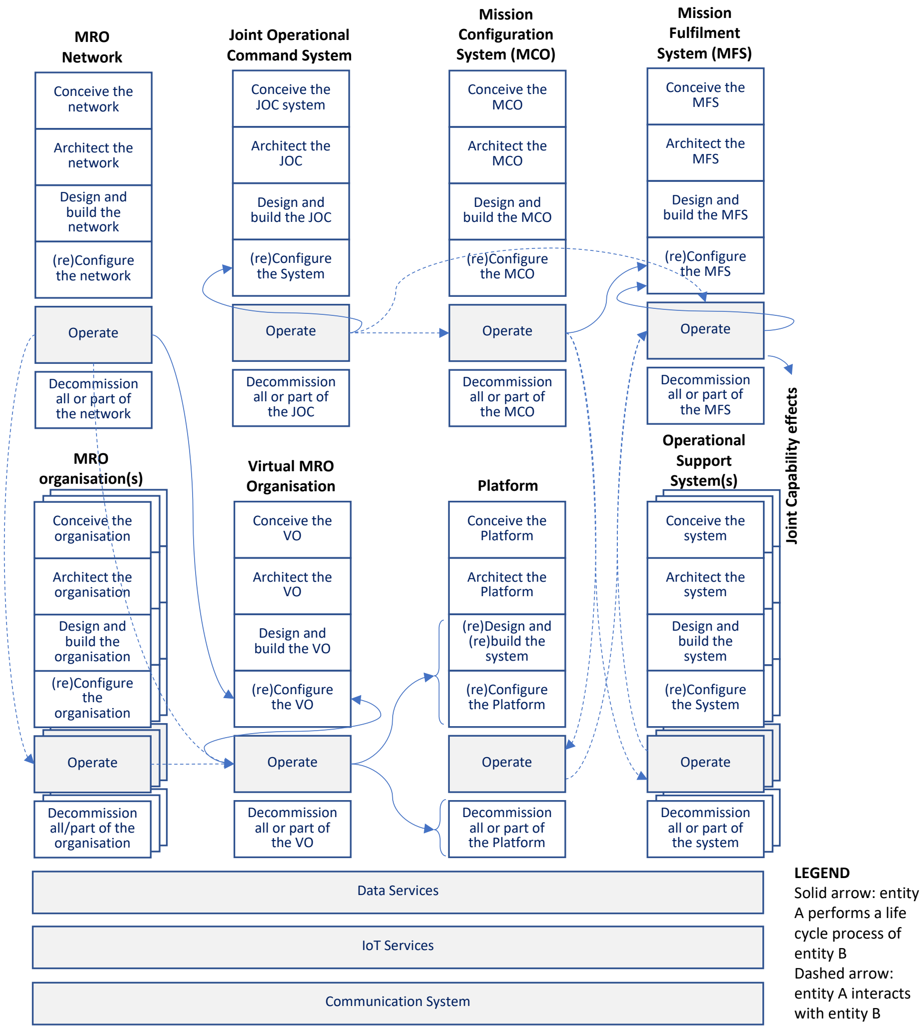

Fig. 2. Life cycle relationships among some major entities of interest in Supportability 4.0

Such a temporal model would then illustrate that there exist two architectural levels of the entities of interest, i.e., a Mission Fulfilment System, the Platform that supports it, and other Operational Support Systems. The two levels are:

- a 'static architecture' that is to be designed to be immutable over a long period of time, and
- a dynamic architecture that is configured in an agile and adaptive way to suit operational situations to match or surpass the pace of operation (*c.f.* OODA loop).

The requirement for this adaptive structure applies both to the mission fulfilment part of the operation, and to its management and control (command and control) part. The latter is often referred to as 'agile C2 (Command and Control)'.

The same adaptive and agile behaviour is required from the MRO (maintenance, repair and overhaul) services, and for this reason the actual service entity can be abstracted into a so-called *Virtual MRO Organisation.* In other words, the dynamism of the support supply chain results from identifying and dynamically creating – and as needed restructuring and optimising – Virtual MRO Organisations for each MRO service instance.

This view is based both on the need for agile MRO services, and on the opportunity afforded by recent technology developments (the ability to sense, distribute and analyse an extensive amount of data across the operation of the platforms, and across the service supply chain).

Notice that in Fig. 2 there exist two loops: i) the MRO Network configures temporary virtual MRO organisations for the service tasks at hand, taking into account both the predictable and ad-hoc needs and optimising for availability and other end user criteria, ii) the MRO virtual organisation adaptively (re)configures itself based on its local control regime, to achieve optimal performance.

The idea of using the virtual enterprise paradigm (Goranson, 1999; Camarinha-Matos, 2004) in establishing agility is not new, this includes the authors having been previously involved in enterprise network- and virtual service enterprise creation (Tølle and Bernus, 2003). However, such previous uses were limited to the creation of virtual enterprises for service *types*. The proposed architectural solution assumes that on-the-fly organisation (and re-organisation) of virtual MRO organisations is possible for each service *instance*, with the consequence of a more resilient and agile solution.

3.2 Supportability 4.0 Architecture

The opportunity hinges on being able to create a common interoperability and data sharing infrastructure environment that complies with the principles above, and is itself supportable in the same way as all other entities of interest should be.

A second but important technology is process optimisation techniques that work on large scale process graphs. Optimal process synthesis was considered for long an intractable combinatorial problem, but recent results (Friedler et al, 1993; Varbanova et al, 2017) using P-graphs allow all viable structures to be generated with comparatively small computing power. This would allow the use of process engineering 'on the fly' in light of the actual support process needs, including for example the real time (re)design of a process that stalled due to unexpected reasons. This dynamic and agile reconfiguration capability is not an available feature of the traditional support supply chain and distinguishes Supportability 4.0 from existing supportability solutions.

The necessary shared services (such as Communication, IoT, Data Services, Analytics, Process Synthesis) must be incorporated into *functionally independent infrastructure layers* that all have the supportability, resilience and agility property. The functional independence requirement is crucial because these layers need to remain technology agnostic (and when technology changes, must be able to be updated without the need to make changes to any of the other layers), thus the entire system of systems remains evolvable.

3.3 Supportability 4.0 Methodology

As the systems in question are self-evolving, there can be no predetermined methodology that is usable for the lengths of time previously described. One can, however, use the previously defined guiding principles and use the concept of a *meta*-methodology to define supportability methodologies for each system, taking into account their life cycles and relationships with other relevant systems. As an example, one such meta-methodology, described by Noran (2004) and used in several case studies comprised the following steps:

1. Determine the relevant entities that support / are to be supported / play other relevant (active or inactive) roles
2. Determine the relations between them *in the context of their life cycles*.
3. Tell the 'life story' of each supported entity, phase by phase. As this is performed, the specific methodology applicable to the system in question is constructed in form of an activity model, which then can be the basis of a transformation plan.

Due to the manner in which this endeavour is performed, the resulting methodology takes into account the life cycle of the systems involved and as such is compatible with the' dynamic architecture' principle stated in Section 3.1.

Below, the authors shall attempt to exemplify the application of the above-mentioned meta-methodology to elaborate a specific method to ensure sustainable, life cycle-aware supportability of the systems involved in the creation and operation of joint capabilities (applicable e.g. in Defence (AusDoD, 2010)).

Step 1 Identifies the entities of interest and their inclusion in the model is justified in light of their relevance to the project at hand. In this case, this step has already been performed in Section 1.3.

Step 2 comprises detailing, re-assessing and graphically representing the entities identified in Step 1. This step also comprises the identification of the relations between the entities (whether changing, providing support or otherwise interacting with each other) in the context of their life cycles. Figure 2 graphically represents the result of applying the second step.

Step 3 comprises telling the story of the entity in question phase by phase, utilising the previously identified and refined set of entities and their relationships. In this case, for example one can follow the formation and operation of an MRO VO by an MRO network, dynamically (re)composing the services of various network member organisations.

This VO then performs the MRO services according to the platform's needs. Note that the mentioned dynamic process configuration solution is part of the MRO Network management, which relies on information that can be extracted from the Data Services (see below), using appropriate filters and analytics.

The Mission Configuration System is akin to a project of short duration created by the Joint Operational Command System, and assembles a Mission Fulfilment System (that includes mission command) using the Platform and other Operational Support Systems (including other Missions): thus, it is in fact the Mission Fulfillment System that creates Joint Capability Effects. The arrow from the MFS operation to MFS (re)configuration represents the fact that mission command may have to change *mode* (which is the essence of agile C2).

The distributed Data access / Data services layer is both a source and repository of sensor data, and relies on the IoT services layer to access the across-the-supply-chain sensor network, which in turn rely on the across-the-chain communication system.

In reality these layers have similar life cycles to the rest of the systems thus their supportability is of equal concern. Traditionally so-called data lakes bundle data storage, access and analytics functions, but for longevity and in light of possible future development in artificial intelligence the analytics and machine learning functions are better kept separate (possibly as part of the entities served).

Another strong logical argument for such separation is the strengthening trend expressed in the terms 'sensing enterprise', 'digital twin', etc. – these represent architectural aspirations to make each of these entities situation aware. Situation awareness (Endsley, 1995) has long been acknowledged as the cornerstone of the 'OODA loop' and has been studied in detail in the C2 literature, including recently as distributed situation awareness (DSA) as it arises when multiple agents / entities are involved in some joint action

(Stanton, 2016; Goranson and Cardier, 2013; Bernus and Noran, 2017).

A logical step beyond current DSA is distributed *self*-awareness, whereupon the system in question is not only aware of the situation in which it is acting, but is also aware of the system's self. While individual agents that are part of some entity (such the human actors in a mission fulfilment system) are self aware, such self awareness does not necessarily arise on the system level.

From the resilience and support point of view attaining this system level self-awareness is a very desirable future development, because the system must be aware of the changes of its own capability in light of external or internal events (such as damage or other events). For example, each agent in the mission fulfilment system may perceive local events that seem 'business as usual', but there may not exist an emerging awareness of the fact that the system is *in the process of being compromised* by the actions of some external (or internal) agent.

The only thing that is constant in this picture is a high-level architecture that allows the identification of life cycle activities to be taken care of for long term supportability, while each entity's detailed design and configuration can change in an agile and dynamic way. Due to the fact that these three layers serve all involved entities, the operational support links have been omitted from Fig. 2. to declutter the figure.

4. CONCLUSION AND WORK FOR THE FUTURE

This paper argued that resilience, supportability and agility are interrelated problems in large scale complex systems, and that for improving the supportability of such systems the scope of enquiry needs to be extended to the systems in the supply chain that support the systems of interest, including operational support- and MRO services. In contrast to traditional supportability methods, the innovation lies in using a systemic approach that considers the effects of supportability and agility requirements on a list of entities involved, not only on the Platforms themselves. These entities include the ones that perform force design, capability acquisition (although for brevity these two were now shown in Fig. 2), operational planning, operations and mission fulfilment, as well as the infrastructure entities that support all of the above.

It was concluded that the life cycle relationships of each of these systems must be studied, understood and systematically assured, and that the list includes systems that provide fundamental underlying layered infrastructure services (supportability challenges arise recursively).

Principles were proposed to guide transformation efforts aimed at attaining better supportability outcomes. Given the legacy of existing systems a serious constraint is that transformation must be carried out in an affordable, risk-mitigated manner and without compromising ongoing operations.

Future work is proposed on two levels:

- Demonstrate in a concrete case the application of the generic architecture outlined in Fig.2, to test the feasibility of the dynamically (re)generated virtual MRO construct. Even though prior examples (developed by the authors) exist, the desired level and speed of dynamic self-configuration has yet to be proven.
- Further develop the theory of self-awareness as an emergent property, in support of attaining higher levels of resilience for large and complex systems.

REFERENCES

AusDoD (2010). The Strategy Framework 2010. Canberra : Australian Department of Defence.

Bernus, P., Goranson, T., Gotze, J., Jensen-Waud, A., Kandjani, H., Molina, A., Noran, O., Rabelo, R.J., Romero, D., Saha, P., Turner, P. (2016) Enterprise engineering and management at the crossroads. *Computers in Industry.* 79 (2016):87-102.

Bernus, P., Noran, O. (2017). Data Rich – but Information Poor. In Camarinha-Matos, Luis M., Afsarmanesh, H., Fornasiero, R. (Eds.) Collaboration in a Data-Rich World. *IFIP Advances in Information and Communication Technology.* (506) pp206-214.

Boehm, B. (2014). -Ilities Tradespace and Affordability Project. Technical Report SERC-2014-TR-039-3. Hoboken, NJ : Systems Engineering Research Center.

Camarinha-Matos, L. M. (Ed.) (2004). Virtual Enterprises and Collaborative Networks. Proc. PROVE04 5th IFIP Working Conf. on Virtual Enterprises. Kluwer.

DAU (2010). Defense Acquisition Guidebook – Ch3. Systems Engineering. Fort Belvoir : Defense Acquisition University.

Dove, R. (1995). Measuring agility: The toll of turmoil. *Production*, 107(1), 12-14.

Dove, R. (2005). Fundamental principles for agile systems engineering. Conf. on Systems Engineering Research, Hoboken , NJ : Stevens Institute of Technology. pp1-12.

Endsley, M.R. (1995). Toward a theory of situation awareness in dynamic systems. *Human Factors*. 37(1):32–64.

Friedler F., Tarjan K., Huang Y.W., Fan L.T., (1993). Graph-theoretic approach to process synthesis: Polynomial algorithm for maximal structure generation. *Computers and Chemical Engineering.* 17(9):929-942.

GAO (2011). Opportunities to Reduce Potential Duplication in Government Programs, Save Tax Dollars, and Enhance Revenue. GAO-11-318SP. Washington, DC : Government Accountability Office.

GAO (2018). Weapon system sustainment. Washington, DC : Government Accountability Office:

Goranson, H.T. (1999). The agile virtual enterprise: Cases, Metrics, Tools. Westport, CT : Quorum Books.

Goranson, H.T., Cardier, B. (2013). A two-sorted logic for structurally modeling systems. *Progress in Biophysics and Molecular Biology*. 113(2013):141-178.

IEC 60812 (2018). Failure modes and effects analysis (FMEA and FMECA). Geneva : IEC.

ISO 15704 (2019). Industrial automation systems - Requirements for enterprise-referencing architectures and methodologies. Geneva : ISO.

ISO 16091 (2018). Space systems — Integrated logistic support. Geneva : ISO.

ISO/IEC/IEEE 15288 (2015). Systems and software engineering -- System life cycle processes. Geneva : ISO.

Jones, J.V. (2006). Integrated Logistics Support Handbook (3rd Edition). New York, NY : McGraw-Hill.

Kandjani, H., Tavana, M., Bernus, P., Nielsen, S. (2014). Co-Evolution Path Model (CePM): Sustaining Enterprises as Complex Systems on the Edge of Chaos. *Cybernetics and Systems: An International Journal.* 45(7):547–567.

NATO (2018). Deployable Communications and Information System (DCIS) Cube Architecting Initiative White Paper. Brussels, Belgium : NATO Communications and Information (NCI) Agency.

Noran (2004). A Meta-methodology for Collaborative Networked Organisations: A Case Study and Reflections, in Knowledge Sharing in the Integrated Enterprise: Interoperability Strategies for the Enterprise Architect. in P. Bernus, M. Fox, and J.B.M. Goossenaerts (Eds.). Dordrecht : Kluwer Academic Publishers: pp117-130.

Osinga, F. (2006). Science, strategy and war: The strategic theory of John Boyd. London, UK : Routledge.

Romero, D., Stahre, J., Wuest, T., Noran, O. S., Bernus, P., Fast-Berglund, Å., Gorecky, D. (2016). Towards an Operator 4.0 Typology: A Human-Centric Perspective on the Fourth Revolution Technologies. In M.I. Dessouky, E.Qi and Z.He (Eds.) Proc. Int. Conf. on Computers & Industrial Engineering (Cie46). Red Hook, NY : Curran Associates, Inc.Vol1. pp608-618.

Stanton, N.A. (2016). Distributed situation awareness. *Theoretical Issues in Ergonomics Science*, 17(1):1-7.

Tølle, M., Bernus, P. (2003). Reference models supporting enterprise networks and virtual enterprises. *Int. J. Networking and Virtual Organisations.* 2(1) Olney Bucks, UK : Inderscience. pp2-15.

UNDRR (2017). UN Office for Disaster Risk Reduction: Terminology. https://www.unisdr.org/terminology (accessed March 2020)

Varbanova, P.S., Friedler, F., Klemeša, J.J. (2017). Process Network Design and Optimisation Using P-graph: the Success, the Challenges and Potential Roadmap. *Chemical Engineering Transactions*. 61:1549-1554