On Systems of Systems Engineering: a Requirements Engineering Perspective and Research Agenda

Cornelius Ncube
Faculty of Engineering and IT
The British University in Dubai, Dubai, UAE
cornelius.ncube@buid.ac.ae

Soo Ling Lim
Braintree Research Lab, Braintree Ltd, UK
Dept. of Computer Science, University College London, UK
s.lim@cs.ucl.ac.uk

Abstract—**The emergence of Systems of Systems (SoSs) and Systems of Systems Engineering (SoSE) is largely driven by global societal needs including energy-water-food nexus, population demographics, global climate, integrated transport, security and social activity. However, due to their scale, structural and functional complexity and emergent properties, these global spanning Cyber-Physical Systems of Systems are becoming increasingly complex and more difficult for current requirements engineering (RE) practices to handle. In this paper, we firstly introduce SoSE as an emerging discipline and key characteristics of SoSs. We then highlight the challenges that the RE discipline must respond to. We discuss some weaknesses of current RE techniques and approaches to cope with the complexity of SoSs. We then argue that there is a need for the global RE community to evolve current RE approaches and to develop new ways of thinking, new RE capabilities and possibly a new RE science as a key mechanism for addressing requirements engineering complexities posed by Systems of Systems. We then outline a requirements engineering perspective and research agenda that identifies 'top-10' research themes informed by a cluster of four Systems of Systems Engineering projects funded by the European Commission's Horizon 2020 research programme.**

Keywords—Systems of Systems, Systems of Systems Engineering, The Fourth Industrial Revolution, Smart Cities, Internet of Things, Cyber-Physical Systems

I. Introduction

Many societal challenges of the 21st Century, such as energy-water-food nexus, global climate change, health, sustainable transport, integrated economy, and coping with the challenge of an ageing population are all highly interconnected and global in nature [1]. A recent United Nations study predicts that, by 2050 nearly 70% of the world's population will be concentrated in urban centres [2] thereby transforming major global cities into intelligent Smart Cities [3]. Due to their enormous scale and complexity, these challenges are multi-dimensional and borderless [1]. Consequently, they are beyond the capacity of any one discipline, sector, country, region to handle or one aspect of policy to address [4][5]. The challenges laid down by the multidimensionality of these global problems including the interconnectedness of decisions across different policy sectors and the emergence of technological-enabled global platforms such as The Fourth Industrial Revolution [6] require new ways of thinking and new Systems Engineering approaches [4]. Addressing these challenges will require systems that work at completely different scales and at completely different constraints than today's systems and will require a major departure from traditional requirements engineering.

A. *Systems of Systems Engineering – an Emerging Discipline*

The emergence of Systems of Systems (SoSs) and Systems of Systems Engineering (SoSE) hence presents a potential for solving many of these challenges [7][8]. Unlike traditional systems engineering which focuses on building a single 'right' system, SoSE focuses on selecting the right combination of constituent systems and their interactions to satisfy a set of frequently changing requirements [9][10]. Hence, while current Requirements Engineering (RE) practices have coped well in dealing with non-SoSs, they are unlikely to result in the same level of success in the dynamic SoSE problem domain.

In this paper, we argue that there is a need for a change in mind-set within the RE community to developing new approaches and to evolve existing RE capabilities as a key mechanism for coping with the complexity of SoSs. We propose *a requirements engineering perspective and research agenda for SoSE* as a first attempt to begin to define a shared international RE vision that will up-scale current approaches and address the development of new theories, techniques, processes, methods and tools for RE practice for SoS.

II. Difference between Systems of Systems and Large Monolithic Complex Systems

There are significant differences between large complex systems (*such as an aircraft*) and SoSs (*such as a dynamic Real-Time Smart City*). SoSs are distinguished from very large but monolithic complex systems by: (a) the *operational independence* of their constituent systems, i.e., each constituent system can operate independently and is capable of achieving its own goals in the absence of the other constituent systems; (b) *managerial independence* of their constituent systems, i.e., the constituent systems are managed independently and can be added or removed from the SoS; (c) the *evolutionary independence* of their constituent systems in which functions and purposes are added, removed or modified as needed; (d) their *emergent behaviours* that cannot be localised to any constituent system, and (e) a *geographical distribution* that limits the interaction of the constituent systems to information exchange [8].

Further, the *operational, managerial* and *evolutionary independence* of the constituent systems are the *three principal distinguishing characteristics* between large complex systems and systems-of-systems. A complex system that does not exhibit these three characteristics is therefore *not considered* a system-of-systems regardless of its complexity or the geographical distribution of its constituent systems [8].

A. *Types of Systems of Systems*

Systems of Systems are further categorised as either [9][10]: **(a) *Directed*,** in which the SoSs is created and managed to fulfil specific purposes and the constituent systems are subordinated to the SoS (Fig. 1a). The constituent systems maintain an ability to operate independently, but their normal operational mode is subordinated to the central managed purpose. *NexGen – US Air Traffic Management System is an example.*

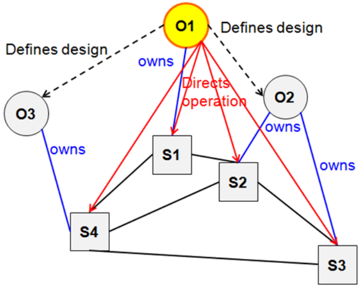

Fig. 1a. In a Directed SoS, operators O2 and O3 accept direction from O1 in terms of the requirements, specification and operation of the systems they own (O2 owns systems S2 and S3; O3 owns S4). This type of SoS is highly controlled by the central managing entity (O1).

(b) *Acknowledged*, in which there are recognised objectives, a designated manager, and resources for the SoS (Fig. 1b). However, the constituent systems retain their independent ownership, objectives, funding, development and sustainment approaches. Changes in the systems are based on cooperative agreements between the SoSs and its constituent systems. *A Smart City is an example of this SoS-type.*

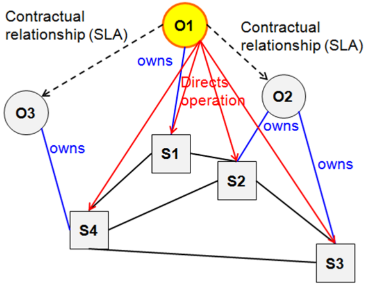
Contractual relationship (SLA)
O1
Contractual relationship (SLA)
owns
Directs operation
O2
O3
owns
owns
owns
S1
S2
S4
S3

Fig. 1b. In an Acknowledged SoS, O1 directs choice of systems and operation; O2 and O3 have a contractual relationship with O1. In this case, the central managing entity (O1), e.g., the Local Government, defines SoS-level requirements but has less control over the constituent systems owned by other city stakeholders O2 and O3 (S2, S3, S4) and must rely more on influence.

(c) *Collaborative*, in which the constituent systems interact more or less voluntarily in an 'agnostic' way to fulfil an agreed-upon central purposes (Fig. 1c). This class of SoSs is best suited to those SoSs where the constituent systems are 'owned' by different organisations, all of which are in fairly equal positions and there is no dominant organisation. *The Global Financial System is an example of this SoS-type.*

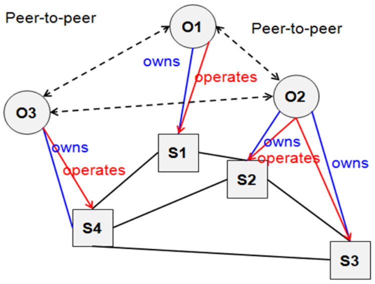

Fig. 1c. In a Collaborative SoS, there is mutual agreement to collaborate usually covered by agreements of some form, but there is no overall managing entity that defines SoS requirements; systems owners (O1, O2, O3) operate their own systems and collaborate with others to realise some shared benefit.

(d) *Virtual*, in which there is no central management authority and a centrally agreed upon purpose for the SoSs (Fig. 1d). Large-scale behaviours emerge but this type of SoSs relies on relatively invisible mechanisms to maintain it. *A Crisis Management System is an example of this SoS-type.*

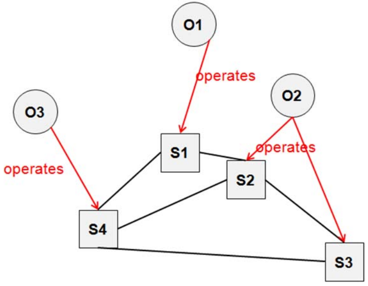

Fig. 1d. In a Virtual SoS, owners (O1, O2, O3) access other systems through their own systems in order to realise individually sought benefits, though high level emergent behaviour may still occur. There is no overall goal, no central management and interoperation is achieved by recognised protocols, or standards, not through individual agreements between pairs of systems.

A note of caution is however necessary for this categorisation. While the four types of SoSs are mostly well-distinguished from each other, in practice there are a few cases in which a SoSs can be categorised as belonging wholly to one type or another. More often, SoSs have very wide-ranging boundaries, therefore different parts of the SoSs can be fitted into different parts of this classification types. Also, it is frequently the case that individual constituent systems may belong to more than one SoSs separately or simultaneously as shown in Fig. 2. Key requirements engineering challenges within this context include: *how to specify SoS requirements, how to predict SoS performance, how SoS requirements affect constituent systems, how requirements for the overall SoS are severely constrained by the existing constituent systems and how to ensure that the SoS achieves its overall purpose.*

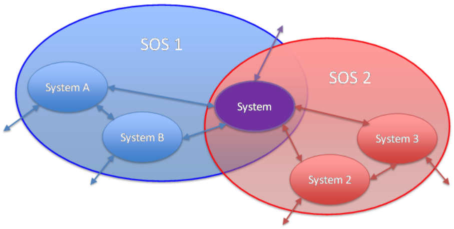

Fig. 2. Constituent systems can participate in multiple SOS.

III. Why Requirements Engineering for Systems of Systems is Challenging

In this section we first highlight why RE for SoSs is challenging. We then highlight some of the key weaknesses of the current RE approaches in addressing these challenges.

A. *RE Challenges in the Context of Systems-of-Systems*

As shown in Table 1, unlike in traditional systems engineering, in SoSE the requirements engineering approach undertaken depends of the type of the SoSs under consideration. Therefore, identifying a SoSs type at the conception stage is key as it influences the entire requirements engineering process. However, given the long lifecycles of most SoSs in relation to the lifecycles of their constituent systems and the changing nature of societal demands upon them, the requirements that a SoSs will need to fulfil during its lifetime are very unlikely to be known at initial requirements engineering phase.

Therefore, a major RE challenge is to define requirements of a system of systems whose lifespan may exceed that of its constituent systems [14]. As SoSs functionality cuts across many constituent systems, requirements engineers face the challenge of allocating SoSs requirements across multiple constituent systems resulting in duplication of functionality. Indeed, as shown in Fig. 2, in a SoSs context, requirements redundancy across individual constituent systems may be acceptable, desirable or even necessary due to the individual roles that constituent systems play apart from the SoSs.

Other added RE challenges in the SoSE context include: a complex stakeholder environment; designing the SoSs in the face of uncertainty; designing for untrustworthy SoSs; designing SoSs that can cope with partial failures; rapidly evolving environments; multi-domains; varied operational context, multi-paradigm shifts in the life of the SoS; etc.

Another, but distinct area in which many of the RE research challenges are similar to SoSs is the Ultra-Large-Scale Systems (ULSS) [58]. These are software systems of unprecedented code size. It is their size and their disorganised complexity that makes traditional centralised requirements engineering approaches inadequate.

To address these challenges, we argue that new requirements engineering approaches need to be developed together with 'scaling-up' existing techniques, tools and best practices to engineer ever-running systems of systems.

B. *Weaknesses of the Current RE Approaches*

The prevailing assumptions of the current RE approaches is that everything flows from the requirements [14]. This logic requires that a good model of the system with fixed requirements and clear understanding of system interactions exist [15]. The assumption is that the system problem is well bounded, the environment is relatively stable and that there is sufficient knowledge to define systems attributes [14]. However, the scale and complexity of SoSs makes it "impossible to acquire all the knowledge to maintain a complete understanding of the SoS" in a single frame of vision in the face of constantly evolving and changing requirements [16]. Also, traditional requirements engineering approaches successively decompose a system "top down from user requirements through to system solution components [16].

However, in SoSE reductionist approaches that require that "a problem be properly defined, the system boundaries established and fixed is not supported" [15]. As such, current 'reductionist' requirements engineering approaches may not either be adequate or appropriate to cope with the characteristics of SoSs that exist in 'conditions of emergence, ambiguity, and uncertainty' [14]. We briefly highlight weaknesses in some of the core traditional requirements engineering approaches next.

Requirements Engineering Process – a traditional requirements engineering process encompasses systematic and repeatable generic activities of requirements elicitation, analysis, specification, verification and management [16]. These activities imply that all requirements need to be identified and verified upfront. However, in a dynamic SoSE environment, the static requirements engineering process is guaranteed to "fail".

Also, in SoSE, the requirements engineering process is impacted by the SoS-type and specific SoS characteristics. For example, in the Acknowledged and Collaborative SoS-types, due to operational and managerial independence of their constituent systems, requirements engineers face the complexity of developing requirements for the SoSs that will be met by the capabilities of independently operated and managed constituent systems whose goals may not be necessarily aligned with those of the SoSs. It is therefore unlikely that a static, leaner process will be applicable these SoS-types.

Use Case-Driven RE Approach – use case modelling provides a pictorial view of how a system will be used by its various stakeholders [17]. The underlying assumption is that all stakeholders and how they will interact with the system can be known in advance. However, Systems-of-Systems are characterised by a complex stakeholder environment with stakeholders at the SoS-level and at individual constituent systems that have different usage needs. This would result in an unmanageable combinatorial explosion in the number of different cases that would need to be considered. Therefore, use case-driven requirements engineering approach might be applicable to the Directed SoS, with some difficulties, though, but it would not be suitable for the other SoSs types.

Goal Modelling Techniques – modelling systems goals and its required services is a core activity in traditional requirements engineering. The underlying philosophy of this technique suggests that systems goals are fixed. However, in a dynamic SoSE problem domain, there is need to identify SoSs level goals and individual constituent systems goals which may be in conflict. As shown in Fig. 1, in a Directed SoSs, the constituent systems are subordinated to the overall SoSs and are contractually obligated to fulfil the goals of the SoS. In this case current goal modelling techniques may be applied. In the Acknowledged and Collaborative SoS-types, constituent systems are independently operated and managed and have their own goals that might not be aligned with the overall SoSs goals. The constituent systems are not obligated to fulfil the SoSs goals and can even refuse the demands of the SoS (e.g., NATO). In a Virtual SoS-type, its goals cannot be elicited, modelled and assigned to agents in advance. Therefore current RE goal modelling approaches such as i* [18], KAOS [19], Tropos [20] and NFR Framework [21] will need to be scaled-up or evolved in order to cope with specific SoS-type and their unique characteristics.

Requirements Acquisition Techniques – currently, there are many acquisition techniques of which some are [22]:

Brainstorming – in which the requirements engineer asks a group of stakeholders to generate as many requirements ideas as possible. In the context of SoSE, brainstorming might be applicable to a Directed SoS but would be difficult to apply to the Acknowledged SoS due to the complexity of the stakeholder environment. In a Collaborative SoS where there is no central authority to identify key stakeholders, brainstorming would not be applicable. In a Virtual SoS brainstorming can be opportunistically applied once the system is in operation in response to specific needs and the dynamic environment.

Observation – in which the requirements engineer observe stakeholders' actual practices in the domain. This technique is not applicable to almost all SoS-types as it requires access to an existing working system. Also, many SoS are abstract, so it will be challenging to observe a system that may not be realisable in the physical domain. However, for the Directed SoS, it is 'theoretically' possible to apply this technique at the constituent systems-level but not at the SoS-level.

Unstructured Interviews – in which the requirements engineer asks stakeholders about the topic without a prepared list of questions and **Structured Interviews** – in which requirements engineer asks stakeholders a list of prepared questions. These techniques rely on gathering a group of stakeholders together in a controlled environment. They assume that stakeholders have complete knowledge to be able to define the SoS attributes. However, due to human limitations, the scale and complexity of SoSs, makes it impossible for any individual stakeholder to acquire all the knowledge and maintain a complete understanding of the system in the face of constantly evolving requirements and changing environment. Therefore, these techniques cannot be applied 'as-is' to almost all SoS types.

Prototyping – in which stakeholders are asked to comment on a physical working model of the desired system. Prototyping is an important part in traditional requirements engineering, but its meaning in SoSs is significantly compromised. In the context of SoSE, the SoS-types pose significant challenges. Even though it is theoretically possible to prototype a Directed or an Acknowledged SoSs, their scale and complexity makes it harder to prototype in the ordinary meaning. The characteristics of the Collaborative SoS make it very difficult to see how prototyping as is currently practiced can apply. As for the virtual-SoS, it would be impractical to generate a physical prototype of a system that does not yet exist.

Requirements Measurement and Metrics – currently, a range of requirements metrics such as volatility, traceability, size and completeness are used to measure requirements. However, in the SoSE problem domain, these metrics would be more complex to define and collect at the SoS-level than in a single system context. For example, the requirements volatility metric measures the degree to which requirements change over time and the reasons for change. While this metrics can be applicable in the Directed SoS-type it would be difficult to apply on the Acknowledged and Collaborative SoS-types due to the operational, managerial and evolutionary independence of the constituent systems.

Traceability metrics measure the ability to follow the 'life of a requirement' in both a forward and backward direction from its origin [23]. However, in many SoSs, it is frequently the case that individual constituent systems may belong to more than one SoSs simultaneously that evolve separately. The operational and managerial independence of the constituent systems and the evolutionary independence of both the constituent systems and the overall SoS makes collecting requirements traceability metrics as is currently practiced inapplicable in this context.

Security Requirements Engineering Frameworks – a number of security requirements engineering frameworks currently exist such as those proposed by Haley et al. [24], Liu et al. [25], Mayer et al. [26] and Giorgini et al. [27]. A common theme among all these security requirements engineering frameworks is their focus on a '*perimeter-centric security model'* that aims to restrict security threats from breaching the perimeter [28]. However, in the SoSE context, security involves not only ensuring that each constituent system is secure but also ensuring that the overall composed SoS is also secure. An added challenge is that in SoSE, security requirements engineering is significantly impacted by the SoS-type and the specific characteristics of SoSs. Hence a perimeter centric security model would be difficult to apply. The SoS geographical distribution characteristics also pose significant challenges due to multiple interactions across multiple infrastructures, domains, policies and regulations.

Emergence Behaviour Requirements – all complex systems exhibit emergent properties that may or may not be predictable. Page [29] identifies three types of emergence behaviours: *Simple emergence behaviour* that occurs in non-complex ordered systems. This is the only type of emergence behaviour that can be predicted; *Weak emergence behaviour* which is expected emergence behaviour that *is desired or allowed* for in the system structure. However, this type of emergence behaviour cannot be predicted from the knowledge of the characteristics of the individual constituent systems; *Strong emergence behaviour* which is unexpected emergence behaviour that cannot be anticipated during the RE phase. This type of emergence is only evident during system failure and cannot be attributed to any particular constituent system(s). Unpredictable strong emergent behaviour, particularly undesirable emergent behaviour, is one of the fundamental challenges in requirements engineering for SoSs [30]. However, upon a review of the requirements engineering literature, we establish that no significant effort has been expedited towards addressing specific requirements engineering emergence behaviour challenges. Indeed, we establish that there are no current requirements engineering approaches or techniques that specifically address emergence behaviour requirements.

C. Brief Summary

We have highlighted some key weaknesses in current requirements engineering approaches in addressing the challenges posed by SoSE and Systems of Systems. We believe that most current approaches are ill-suited for addressing the complexity and interconnectedness nature of SoSs requirements. They oversimplify reality through a reductionist approach, and mostly follow predictable linear process. Many of them simply *do not* scale-up to complex SoSs.

However, a note of caution is necessary. The highlighted weaknesses are not an attempt to dismiss all traditional requirements engineering approaches as wholly inapplicable in SoSE. Rather, we suggest that because of the different SoSE problem domain, the different types of SoSs that require specific attention, the specific SoS characteristics of operational, managerial and evolutionary independence of constituent systems and strong emergent behaviour properties, current requirements engineering approaches need to be revisited, evolved or scale-up in order to cope with these unique challenges. Further, the arguments presented above should be viewed as a basis for suggesting that a paradigm shift is required and necessary to develop new requirements engineering approaches for Systems of Systems. We do not believe that requirements engineering approaches that have been successful in traditional systems engineering would *'enjoy the same level of success'* if practiced in their current form to the more *'holistic, emergent, and uncertain'* problem of SoSE [9]. It is these arguments we believe question the viability of the current RE approaches that necessitates the proposed research agenda that is described in the remainder of this paper. Table 1 shows that in the SoSE problem domain, the requirements engineering approach taken depends on the type of the SoS being considered.

TABLE I. A TAXONOMY OF RE APPROACH IN SOSE CONTEXT

SoS Type	***Example SoSs***	***RE Approach***
Directed	NexGen – US Air Traffic Management Systems. These are well-specified and stable SoS that are developed to answer specified needs	A single, large, dominant organisation is obliged to take on a requirements engineering leadership role, (e.g., FAA for NexGen) and defines both the SoS-Level requirements and Constituent Systems Level requirements. The central SoS authority also controls, mandates and directs the lifecycles of the Constituent Systems. Both the SoS-level and CS-level requirements evolution are controlled and coordinated by the central authority. In this type of SoS classical RE approaches, methods, techniques and tools may apply.
Acknowledged	NATO Alliance, Smart City. SESAR-Single European Sky (EU),	The Constituent Systems are owned and operated by different organisations, e.g., NATO Members, However, SoS-level requirements engineering is performed by central authority (e.g., NATO, Smart City) and addresses requirements across the SoS. RE for each Constituent System is performed independently and addresses requirements from their owner's perspective with independent lifecycles. Because RE is performed independently by the SoS owner and the individual Constituent Systems with minimal collaboration, classical requirements engineering approaches, may not apply without significant scaling-up.
Collaborative	The Global Financial System. Potentially competing organizations come together to agree standards, protocols for processes and products within the industry.	In this SoS-type, there is no single dominant organisation which performs the leadership role for SoS-level requirements engineering. Requirements Engineering is performed by each Constituent Systems independently. However, since no one group owns the SoS, there is no obvious group that owns its requirements. Because of the operational and managerial independence of the constituent systems, classical current requirements engineering approaches may not apply.
Virtual	A Crisis Management System, The Global Information Grid. For example, the Crisis Management SoSs is only created once the crisis response is in progress.	Virtual SoSs lack any central purpose and are generally, at most, informally guided by the users. The SoS purposes are dynamic and requirements change frequently. There is no central requirement engineering authority. There is informal and irregular, if at all, requirements engineering at constituent systems level. How the SoS will be composed and the participating constituent systems cannot be known in advance. Current requirements engineering approaches do not apply.

IV. HOW THE RESEARCH THEMES WERE DEVELOPED

Figure 3 shows the methodology used to drive the RE-specific research themes. In preparation for Horizon 2020 programme [5], the European Commission (EC) funded a cluster of four related SoSE projects contributing to the creation of the EC's SoSE Strategic Research Agenda [31][32][33][34]. Over a period of 17 months, more than 150 experts from the SoSE International Expert Community from industry, government, policy-making and academic communities together with research funding agencies in both Europe and the US; the European Commission, the IEEE Technical Committee on Systems of Systems, the INCOSE Working Group on Systems of Systems [35] contributed to the empirical development, prioritisation and validation of the SoSE Strategic Research Agenda in an online survey and a series of workshops.

The RE-specific research themes that are proposed in this paper have been generated as a result of an integrated synthesis of the SoSE State-of-the-Art, the EC SoSE Strategic Research Agenda, State of current RE approaches, Weaknesses of current RE approaches, SoS societal drivers, Visionary Scenarios, RE challenges in the context of SoSs (as described in Section 3), a panel session at RE13, our experiences in the four SoSE projects described in [31][32][33][34], literature review, desktop research and editing activities. Table 2 gives a brief description of each research theme and Section 5 describes each research theme in detail.

A note of caution is necessary: The main goal of this paper is to propose a roadmap to provide strategic directions for the future of requirements engineering research for the next 25 years. The 10 research themes that are presented in this paper are not an exhaustive list. Rather they are a starting point to guide the research directions of the international requirements engineering community at large. They constitute a shared vision that we hope will be able to inspire collaborative and interdisciplinary research between academia, industry, civil society and government. We hope that the RE community will find certain research themes resonant with their own research *interest and will proceed accordingly to finding solutions to the requirements engineering challenges posed by SoSE.*

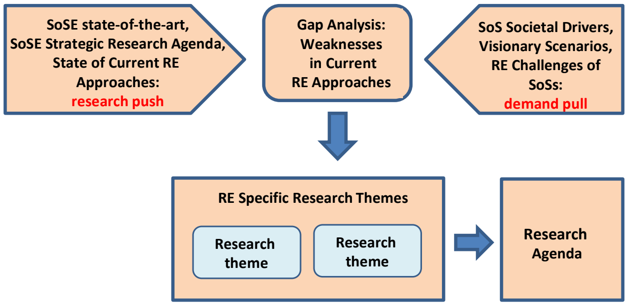

Fig. 3. Strategies adopted for generating the research themes.

V. The Research Agenda: The Future of Requirements Engineering in the Next 25 Years

The intention of this paper is to provide a research agenda that identifies 10 requirements engineering research themes that we believe attention should be devoted by both practitioners, who wish to "reengineer" their RE processes, and academics seeking intellectual challenges. Each research theme is presented next in ***no priority*** order:

Theme 1: Requirements Engineering Approaches for Defining SoSs Emergence Behaviour Requirements

The task of SoSE is, in essence, the task of engineering for emergence behaviour. These global behaviours emerge from the cumulative actions and interoperations of the constituent systems that are propagated throughout the system of systems. Emergent behaviour arises from influence requirements through two primary mechanisms: *emergent composition* and *cascade effects* [37]. *Epidemics* are special forms of cascade effects that are able to influence a large number of constituent systems without having to direct information to particular systems [37][38].

However, we currently do not understand enough about *influence requirements* by which constituent systems interact to produce emergent behaviour. Furthermore, due to their inherent nature, Systems of Systems always fail - sometimes partially and sometimes more generally. To minimise the effects of failures and to maintain the integrity of the overall SoS, a failure that affects only one constituent system should not affect the entire SoS [39]. However, we currently do not understand enough about *failure requirements.* We also do not understand enough about *requirements for recovery* that would allow the SoSs 'to restart with the minimum of disruptions' [15].

Also, the specific characteristics of SoS and the type of SoS makes developing requirements for predicting emergent behaviour an added challenge for requirements engineering. For example, in the case of a Directed SoS, both good and bad emergence behaviour requirements can be identified and designed in or tested out of the SoS as appropriate. The SoS and the individual constituent systems can be tested to see if they exhibit the emergent behaviours. In this case, some current requirements engineering approaches can be applied to identify

TABLE II. Brief Descriptions of the 10 Research Themes

Research Theme	Brief Description
RE Approaches for Defining SoSs Emergence Behaviour Requirements	The unpredictability of emergent behaviour is one of the fundamental challenges in RE for SoS. This theme addresses the need for specific requirements engineering approaches for identifying, analysing and understanding SoS emergence behaviour requirements.
SoSs Security Requirements Engineering Frameworks	This research theme addresses the need for security RE approaches that provide a holistic security view for securing increasingly complex SoSs which are susceptible to constant changes that exposes them to ever new security threats
RE Approaches for Defining & Evolving SoSs Architecture Requirements	This theme addresses the need for requirements engineering techniques and tools for defining requirements for composing, scalable SoS dynamic architectures as the SoS expands in scope, functional and structural complexity
Evaluation Metrics of SoS Requirements	The theme addresses needed research to develop requirements engineering approaches for combining requirements metrics to evaluate the performance of a SoS against expected behaviour, desired outcomes, and in comparison with other possible configurations. It also addresses the need for a framework to evaluate how well do constituent systems capabilities meet SoS-level requirements for a given configuration.
Characterisation and Description of SoSs Requirements	As SoSE is inherently multidisciplinary, there is a need for a common way for characterising and defining SoS requirements. This theme is concerned with developing approaches for all disciplines to describe the structure of SoS-level requirements.
Theoretical Foundations for SoSs Requirements Engineering	This theme is concerned with developing a theory for requirements engineering for system of systems or potentially a number of unifying RE theories that are dependent on the SoS-type. This would provide a more structured, theoretical basis to requirements engineering of complex Systems of Systems.
Multi-Level Modelling Techniques for SoSs Requirements	This theme is concerned with the development of both fundamental and pragmatic requirements engineering modelling approaches that reflects the holistic nature of SoS requirements. It is aligned with the theoretical foundations theme and concerns the development of RE modelling approaches that give insight into the fundamental nature of SoS requirements.
SoSs Requirements Measurement Metrics	The different SoS-types, the operational, managerial and evolutionary independence of the constituent systems and the unpredictable emergent behaviours of the SoS makes it difficult to define useful requirements measurements metrics. This theme is concerned with development of requirements engineering techniques and tools for identifying appropriate SoS requirements metrics and associated methods of measuring them.
Virtual Requirements Engineering Environments	The inherent nature Systems of Systems makes them very difficult to analyse through conventional prototyping approaches. This theme is concerned with the of development of continuous *virtual requirements engineering environments* that incorporates requirements modelling, simulation and visualisation tools that will make use of the feedback from the SoS and constituent systems to continuously refine requirements and evolve new system designs.
Tools for SoS Requirements Trade-Off Decisions	This theme is concerned with the development of RE strategies and tools to enable effective trade space decisions for a variety of potential stakeholders to create different SoS versions for a given purpose, and then to choose the 'best' one.

desired emergence behaviour requirements. However, in the Acknowledged and Collaborative SoS- types, the interactions among the constituent systems are likely to result in unexpected emergence behaviours that cannot be anticipated or foreseen. As both the SoS and the individual constituent systems cannot be tested to see if they exhibit emergent behaviours, it is therefore unlikely that current requirements engineering approaches will apply in these SoS-types. In the case of a Virtual SoS-type in which emergence behaviours emerge during operation as a result of either the evolution of the SoS itself or the environment, chaotic or truly unpredictable emergence behaviours are likely for this type of SoS. We do not believe that any current requirements engineering approaches would apply in a virtual SoS-type.

Theme 2: SoSs Security Requirements Engineering Frameworks

For many SoSs, the benefits of interoperation are realised at the expense of introducing new opportunities for security breaches, either through malicious or accidental activities. As constituent systems may enter or leave (and return) while the SoS is in operations, it is therefore vital that they do so with some predictable confidence that the trustworthiness of the overall SoS is not weakened. Therefore, in the SoSE context, security requirements engineering involves not only ensuring that the overall SoSs is secure but also ensuring that each constituent system is also secure. Security requirements must be propagated throughout the entire life cycle of the SoS and its constituent systems which could be decades. However, the *operational, managerial* and *evolutionary independence* of the constituent systems as well as the *geographical distribution* and *emergent properties* of SoSs, a *large and complex attack surface* and *numerous points of attack* increase the complexity of the security requirements landscape.

As discussed in Section 2, a number of security requirements engineering frameworks currently exist. Although we regard these approaches good-enough, their applicability in dynamic SoSE problem domain is called into question as they do not provide a holistic security view that is necessary to secure complex systems-of-systems. They lack mechanisms to determine if the security of one or more constituent systems has been compromised when they join the SoS. What is required is an adaptable, decentralised and distributed security framework that will underpin trust and integrity between the constituent systems and that would evolve as the SoS evolves. Because of its decentralization and distribution, coupled with its inherent combination of consensus mechanism and secure protocols, we believe that a Blockchain-Based Security Requirements Engineering Framework would alleviate some of the concerns of current approaches [40]. A Blockchain-based security requirements engineering framework would integrate a consensus mechanism in the SoSs architecture that would allow a majority or all constituent systems to agree to work together as a group or even survive if some of them have been compromised. The SoS would then leverage Blockchain's smart contract technology to establish agreements between its constituent systems. A trustworthy SoS must be able to cope with one or more of its constituent systems being compromised.

However, in a Blockchain-based security framework, with its consensus mechanism, to compromise the security of the overall SoSs would require compromising the security of all or over half of the constituent systems simultaneously to be successful, which would be nearly impossible to accomplish by attackers [41]. We therefore believe that a Blockchain-based Security Requirements Engineering Framework would not only provide constituent systems protection, but a holistic protection of the overall SoSs. More broadly, we believe that there is a need to develop RE strategies and techniques for analysing how small requirements changes can affect the overall SoS trustworthiness; methods and strategies for decomposing SoS trustworthiness goals into specific security requirements that are allocated to specific constituent systems and verifying the overall SoS trustworthiness properties.

Theme 3: Requirements Engineering Approaches for Defining & Evolving SoSs Architecture Requirements

Architecture requirements are those requirements that have particular attributes that are 'implicitly or explicitly architecturally significant' [42]. An SoS architecture provides an integrated view of the ensemble of constituent systems within the SoS. It does not address the details of the individual constituent systems but rather defines the Concept of Operations detailing the way constituent systems will interoperate to meet SoS requirements [43][44]. However, the requirements engineering task is made more challenging by the fact that in most SoS-types, constituent systems are managerially and operational independent entities that serve other SoS concurrently. It is the responsibility of the requirements engineer to specify a SoSs architecture that meets these requirements and this poses major challenges to current requirements engineering approaches.

Another key requirements engineering challenge is to develop techniques and tools for handling dynamic scalable architectures as the SoS expands in scope, functional and structural complexity. For example, existing approaches such as TOGAF [43], Twin Peaks [45] and Architecture-driven Requirements Engineering [46] may be applicable to the Directed SoS-type and to some extent to the Acknowledged SoS-type. However, the operational and managerial independence of the constituent systems of a Collaborative SoS pose major challenges as they are designed according to their unique architectural style and to serve their own purposes and not to optimise SoSs objectives. The evolutionary independence of the constituent systems and of the SoS simultaneously necessitates a dynamically changing and evolving SoS architectures. Architecture requirements engineering modelling tools and frameworks for understanding architecture emergent behaviour requirements are needed [12]

Specific RE challenges are to develop a better requirements understanding of the relationship of architectural features to SoS behaviours, the assessment and evaluation of architectures requirements, and architecture requirements for a range of SoS stakeholders.

Theme 4: Evaluation Metrics of SoSs Requirements

Understanding the behaviour of the constituent systems and their interactions is important in achieving the resilience of the overall SoS. Many requirements engineering metrics for evaluating requirements for non-SoS currently exist. However, in the SoSE context, the task of evaluating requirements depends on the SoS-type and its specific characteristics. For example, in an Acknowledged and Collaborative SoS-types, the operational and managerial independence of the constituent systems, the individual roles constituent systems play apart from the SoS [9] makes it impossible to apply the current requirements evaluation metrics. In these SoS-types, requirements for constituent systems are specified, evaluated and managed separately by each constituent system owner using their own methods. In a Virtual-SoS, the unpredictable emergent properties make currently existing metrics inapplicable. The major challenge is to define requirements metrics for evaluating the behaviour of the individual constituent systems to identify anomalous behaviour and how a disruption in one constituent system directly impacts the behaviour of other constituent systems and the behaviour of the overall SoS. It is these SoS aspects that suggest new requirements evaluation metrics and strategies that can be applied to a range of SoS types at different levels and operational environments are required [14].

Theme 5: Requirements Engineering Approaches for Characterisation and Description of SoSs Requirements

In traditional systems engineering, requirements are generally specified as either System level requirements which relate to the system as a whole; Use Case level requirements that relate to a particular use case and Action level requirements that relate to an individual action within a use case [17]. Each requirement is described as an 'atomic shall statement'; has a type and a tag that identifies its type; a rationale for the requirement; a unit of measurement criteria and a source who is usually a stakeholder who proposed the requirement [47]. However, this requirement characterisation would be difficult to apply in the SoSE problem domain. SoS-level requirements are not usually described as atomic 'shall statements' but are described as needed capabilities that are addressed using the capabilities of existing constituent systems, i.e., the whole constituent system can be defined as an SoS-level requirement. As the SoSE problem domain is inherently multidisciplinary there is a need for a common way of characterising and describing SoS requirements that is applicable across many disciplines and domains. However, the absence of a well-established base for characterising and describing SoS requirements is an obstacle to effective SoS requirements engineering. Hence addressing this theme will create a shared understanding of SoS requirements concepts and help to move more effectively from traditional requirements engineering paradigm to a SoSE requirements engineering paradigm.

Theme 6: Theoretical Foundations for SoSs Requirements Engineering

Whilst the realities of requirements engineering challenges for Systems of Systems are apparent, a generally agreed theoretical basis upon which to do it is lacking. Current SoS RE approaches such as those proposed by [48][49] are not based on solid theoretical foundations. The NATURE framework [50] provided a set of interacting theories that underpin traditional requirements engineering. The NATURE framework is comprised of the *domain theory* that describes 'the knowledge structures people develop and remember when investigating a problem'; the *requirements process theory* that views the requirements engineering process as a system and provides means for describing both its static structural and dynamic parts; the *knowledge representation theory* that represent requirements in a combination of various forms.

However, even though the NATURE framework provides comprehensive requirements engineering process guidance and tool development, its underlying theories would be a challenge to apply in the SoSE problem. For example, most SoS are multi-domain therefore due their scale and complexity and human cognitive limitations, it would be nearly impossible for any stakeholder to develop complete requirements knowledge of the whole SoSs. Therefore, the domain theory falls short. As already argued in Section 2, leaner requirements engineering processes are not adequate for the SoSE problem, hence the requirements process theory fails. Also, as described in Theme 4, SoS-level requirements are described in terms of needed capabilities, it would be difficult to formally represent them as advocated by the knowledge representation theory. Even though these theories have been applied successfully in traditional requirements engineering, the different SoS-types and their specific characteristics makes their direct applicability in the SoSE problem domain questionable.

New requirements engineering theories or a number of integrated theories that are independent of the SoS-type are required to provide a more structured, theoretical basis to the requirements engineering of complex systems of systems and to form a basis for a broad range of RE methods and tools. These could include new theories or an appropriate organisation of existing theories from natural or biological systems that exhibit complex behaviours or from other disciplines such as systems science, mathematics, human and social sciences.

Theme 7: Multi-level Modelling Techniques for SoSs Requirements

In general, requirements describe a system to be built. They mainly focus on the problem that the envisioned system solves and not on the specific implementation details. Models are the basis of understanding the world as well as for communicating among all stakeholders. A requirements model is a structured representation of requirements, using a specific modelling language which can be either textual or graphical. Currently many requirements modelling approaches exist. The UML Requirements Modelling approach uses Use Cases to model high level requirements, and Class Diagrams to model the system domain [51]. The Systems Modelling Language (SysML) is used for specifying and analysing complex systems [52]. Greenspan [53] proposed the Requirements Modelling Language (RML) which models a system from a business analysis perspective focussing on business goals and objectives.

Berenbach and Gall [54] build on the work of Greenspan [53] and proposed the Unified Requirements Modelling Language (URML) that allows the capturing of essential requirements engineering information in the early phases of systems development. Glinz and Wüest [55] proposed the Very Lightweight Modelling Language (VLML) for capturing early requirements. Holt et al [49] proposed the Model-based requirements engineering for systems of systems approach that is based on the COMPASS project [33]. Yu [18], Dardene et al. [19], and Mylopoulos et al. [21] proposed goal-modelling techniques that are described in Section 2. While these requirements modelling approaches might have been successful in traditional requirements engineering, their applicability in the dynamic SoSE problem domain is not guaranteed to succeed. They are just too functionally oriented and too specific to be able to capture complex dynamic SoS emergent behaviours requirements.

Therefore, to adequately model requirements for SoS requires a combination of modelling approaches from different perspectives, with different views, and with different levels of fidelity that must be integrated into a system of models, i.e., an *open model interoperability framework*. The challenge is therefore for requirements engineering discipline to revisit current modelling techniques or to develop new ones that can model evolving requirements relationships within a SoS and between constituent systems taking into consideration particular SoS-wide dynamic emergent behaviour patterns.

Theme 8: SoSs Requirements Measurement and Metrics

Currently there are no universally agreed sets of metrics for measuring or assessing SoS-level requirements. The operational, management and evolutionary independence of the constituent systems makes it difficult to make useful requirements measurement metrics. While traditional requirements measurement metrics such size, i.e., the number of requirements, quality, atomicity, unambiguity, completeness, traceability etc. can be defined, collected and analysed for the Directed SoS-type, this would be difficult for the Acknowledged, Collaborative and Virtual SoS-types as the required information is handled by multiple owners.

Although there has been major research efforts on requirements traceability [23][56], in the context of SoSs maintaining accurate requirements traceability measurement metrics remains a major challenge [57] due to the centralised requirements management approaches. Research is therefore required to develop approaches for identifying, tracking and maintaining requirements metrics that are relevant for the entire lifespan of the SoS and its constituent systems as they both evolve independently. One possible solution would be to exploit the geographical distribution characteristics of SoS by adopting Distributed Ledger and Blockchain technologies as a framework for tracking and managing requirements traceability metrics [40][41]. In a Blockchain based framework, the requirements database would be maintained as a distributed ledger that is shared across all participating constituent systems and the SoSs. All participating constituent systems and the SoS itself will have their own identical copy of the requirements ledger. Any requirements modifications will be reflected in all copies of the requirements ledger at the same. To enable identifying and tracking requirements metrics through the life-spans of the constituent systems and the SoS, each requirement will be defined in the requirements ledger with a digital passport that proves its authenticity and origin, thereby creating a secure auditable record of its changes. Research is also required to develop security-specific metrics and measures at the SoS-level.

Theme 9: Virtual RE Environments for Prototyping, Simulation and Visualisation of SoSs Requirements

Prototyping is an important part of requirements engineering in traditional systems engineering. However, its meaning is significantly compromised in the SoSE problem domain. The inherent nature of Systems of Systems such as Air Traffic Management Systems, Smart Cities, Industry 4.0 or Crisis Management Systems makes them very difficult to analyse through conventional prototypes due to their scale and complexity. What is required is new research to develop requirements test-beds that would replicate realistic operational behaviours of the SoS in a virtual simulation environment [59][60]. This research should develop methods that would recreate realism of a SoSs while providing flexible trade-off analysis between the SoS architecture, constituents systems and the requirements.

These large-scale *virtual requirements engineering environments* should incorporate requirements modelling, simulation and visualisation tools that would enable virtual interactive requirements analysis of a SoS from different stakeholder-specific viewpoints.

Also, we believe that the only realistic way of addressing the RE challenges outlined in Themes 1, 2, 3 and 4 will be through large scale virtual requirements simulation and visualisation experimentations [61]. For example, using the virtual experimental environment, requirements engineers can build exploratory requirements models that simulate the interactions of constituent systems and the behaviour of the overall SoS to calculate its energy consumption or the risk of a cascading failures. They can also build threat models of a SoS and then virtually attack it to identify security vulnerability requirements. They can even digitally 3D-print the SoS. *However, it should be noted that virtual requirements simulation environments can only demonstrate the presence of desirable emergent behaviour requirements, but are unlikely to demonstrate the absence of undesirable emergent behaviours*.

Theme 10: Tools for SoSs Requirements Trade-Off Decision

Requirements trade-off is a well-established practice in traditional systems engineering [62][63]. However, in the dynamic SoSE problem domain, the task of defining and parameterising the requirements trade space is complicated by factors such as a lack of definition of the SoS, the dynamic nature of the SoS and its environment, the operational and managerial independence of the constituent systems and the unpredictable behaviour properties of the SoS [64]. In the Acknowledged and Collaborative SoS-types, the requirements trade space is very dense, as configuration of different constituent systems may provide a large number of options. Research in this theme should address the development of strategies and tools to enable effective requirements trade-off decisions for a variety of stakeholders.

We currently lack requirements engineering techniques that enable to create different SoS versions for a given purpose, and then let decision-makers choose the *'best one'*. This research should therefore include development of requirements engineering feasibility methods for determining whether the SoS is feasible and for identifying feasible solutions [65].

Trade-offs between SoS architecture, constituent systems and requirements as well as associated spectrum of risks including security are essential during analysis of alternatives for both individual constituent systems and the SoS as a whole. This research should therefore develop requirements sensitivity and uncertainty analysis tools to analyse the SoS from all perspectives including performance, security, resilience and risk through its entire life-cycle and of its constituent systems.

Please Note: As the research themes are not prioritised in order of importance, we believe nothing of value will be gained from grouping them. However, we suggest that the roadmap for addressing each research theme should proceed as follows:

- *Short Term:* innovative research actions, pilot studies and experiments across disciplines, industry take-ups ~ 1-3 years
- *Medium Term:* incremental experimental-driven research building on the short-term actions and approaches needed to be developed ~ 2-4 years
- *Long Term:* more radical research and development needed ~ 4+ years

VI. Timeliness and The Need for a New Research Agenda

In their 2000 Requirements Engineering roadmap, Nuseibeh and Easterbrook [22] highlighted some key open RE research issues for the future. They described core requirements engineering activities and how these can be integrated into a single process. They also discussed methods, elicitation and modelling techniques for the integrated RE process. They concluded by identifying some key challenges for future RE research.

We believe that progress has been made since then in addressing some of these challenges. However, as shown in Table 1, in the dynamic SoSE problem domain, the SoS-type (i.e., *directed, acknowledged, collaborative or virtual*) determines the requirements engineering process to adopt. We therefore do not believe that a single integrated RE process would be applicable to all SoS types. Also, as outlined in Section 2, we believe that most current RE methods, tools, elicitation and modelling techniques would need to be revisited or evolved as proposed in this paper in order to cope with the challenges of SoSs.

In 2007, Cheng and Atlee [66] built on the work of [22] and identified nine RE research hotspots. The identified hotspots included scale, complexity and security for software systems as posing challenges to future RE research. This paper builds on the work of [66] to identify scale, complexity and security as among primary requirements engineering challenges for SoSE.

In 2011, Franch et al. [67] proposed an i*-based framework that incorporates other modelling techniques to overcome some key RE challenges. We believe this work is in the right direction and could result in integrated open goal-modelling frameworks that can cope with SoS challenges if the weaknesses outlined in Section 2 are overcome. In 2011, Milne and Maiden [68] considered the role of power and politics in requirements engineering. We believe that power and politics are a major factor in SoS of societal significance.

As we stated in Section 1, the emergence of SoS and Systems of Systems Engineering is largely driven by global societal needs that cross domains, boundaries, jurisdictions, policy and regulatory authorities. Solutions to these challenges would be national or global scale in scope and interoperating across many national and international critical infrastructures, policies and regulatory authorities. Therefore, politics and power play a key part in requirements engineering for SoS.

Now that we have provided the "*we are here maker*", we believe that the proposed research agenda outlined in this paper provides a 'big arrow' *sign-posting the future* requirements engineering roadmap for the next 25 years

We believe that the way forward should include the active engagement of all communities using the proposed research themes as a starting point. The *RE research community* should develop *rigorous* and *pragmatic theoretical foundations* that would be the basis for requirements engineering methods, techniques and tools for SoSE. *Educational institutions* and *policy makers* should prepare the *requirements engineer of the future* by investing in and building key skills and competencies that are grounded on solid *theoretical foundations*. The *industrial tool vendors* should evolve current RE tools and methods and build new ones based on rigorous and pragmatic theoretical foundations from the research community.

Acknowledgment

Thanks to Ian Sommerville (St Andrews University) for the feedback on the earlier draft of this paper and to Anthony Finkelstein (UCL), Nancy Mead (SEI) Judith Dahmann (US DoD & MITRE Corporation) and Alan Harding (BAE Systems) for their contribution in refining the research themes. This work was funded by the European Commission under the Horizon 2020 programme.

References

[1] S. Dutta and B. Bilbao-Osorio, *The Global Information Technology report: Living in a Hyperconnected World*. World Economic Forum, 2012.

[2] United Nations, Department of Economic and Social Affairs, Population Division. *World Urbanization Prospects: The 2014 Revision*, Highlights (ST/ESA/SER.A/352), 2014.

[3] Securing Smart Cities. [Online]. Available: http://securingsmartcities.org/. [Accessed: 29 December 2017].

[4] Global Systems Science. [Online]. Available: http://www.gsd.eu. [Accessed: 21 January 2018].

[5] Horizon 2020, *The EU Framework Programme for Research and Innovation*. [Online]. Available: http://ec.europa.eu/programmes/horizon2020/. [Accessed: 21 January 2018].

[6] K. Schwab, *The Fourth Industrial Revolution*. Crown Business, 2017.

[7] M. Jamshidi, *Systems of Systems Engineering – Innovations for the 21st Century*. Hoboken, NJ: John Wiley & Sons, 2009.

[8] M.W. Maier, “Architecting principles for systems-of-systems”, *Systems Engineering*, vol. 1, no. 4, pp. 267-284, 1998.

[9] J.S. Dahmann, and K.J. Baldwin, “Understanding the current state of US defense systems of systems and the implications for systems engineering”, 2nd Annual IEEE Systems Conference, Montreal, 2008.

[10] M. Henshaw, C. Siemieniuch, M. Sinclair, V. Barot, S. Henson, C. Ncube, S.L. Lim, H. Dogan, M. Jamshidi, and D. DeLaurentis, *The Systems of Systems Engineering Strategic Research Agenda*. 2013.

[11] Pétrissans, A., S. Krawczyk, L. Veronesi, G. Cattaneo, N. Feeney and C. Meunier (2012). Design of future Embedded Systems toward system of systems - trends and challenges, International Data Corporation (IDC) Analyze 2012

[12] Office of the Deputy Under Secretary of Defense for Acquisition and Technology, Systems and Software Engineering. *Systems Engineering Guide for Systems of Systems*, Version 1.0. Washington, DC: ODUSD(A&T)SSE, 2008.

[13] B. Collins, “Needs and Challenges for Systems Thinking”, Presentation at the Department for Business Innovation and Skills, 2011.

[14] C.B. Keating, J.J. Padilla, and K. Adams, “System of systems engineering requirements: challenges and guidelines” *Engineering Management Journal*, vol. 20, no.4, pp. 24-31, 2008.

[15] I. Sommerville, “Design for recovery: challenges for developing complex systems of systems”, 7th IEEE International Conference on Composition-Based Software Systems (ICCBSS), Madrid, 2008.

[16] I. Sommerville and P. Sawyer, *Requirements Engineering: A Good Practice Guide*. New York, NY: John Wiley & Sons, 1997.

[17] I. Alexander and N. Maiden, *Scenarios, Stories, Use Cases: Through the Systems Development Life-Cycle*. John Wiley, 2004.

[18] E. Yu, “Towards modelling and reasoning support for early-phase requirement engineering”. *IEEE International Symposium on Requirements Engineering*. pp. 226-235, 1997.

[19] A. Dardenne, A. Lamsweerde, and S. Fickas, “Goal-oriented Requirements Acquisition”. *Science of Computer Programming*. vol. 20, nos. 1-2, pp. 3-30, 1993.

[20] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos, “Tropos: an agent-oriented software development methodology”. *Autonomous Agents and Multi-Agent Systems*, vol. 8 no. 3, pp. 203-236. 2004.

[21] J. Mylopoulos and L. Chung, “From object-oriented to goal-oriented requirements analysis.” *Communications of the ACM*, vol. 42 no .1, pp. 31-37, 1999.

[22] B. Nuseibeh and S. Easterbrook, “Requirements engineering: a roadmap”, In *Proceedings of the IEEE International Conference on Software Engineering (ICSE)*, pp. 35-46, 2000.

[23] O. Gotel and A. Finkelstein, “An analysis of the requirements traceability problem”, In *Proceedings of the First International Conference on Requirements Engineering*, pp. 94-101, 1994.

[24] C. Haley, R. Laney, J. Moffett, and B. Nuseibeh, “Security requirements engineering: a framework for representation and analysis”, *IEEE Transactions on Software Engineering*, vol. 34 no. 1, pp.133-153. 2008.

[25] L. Liu, E. Yu, and J. Mylopoulos, “Analyzing security Requirements as Relationships among Strategic Actors. In 2nd Symposium on RE for Information Security (SREIS), 2002.

[26] N. Mayer, A. Rifaut, and E. Dubois. Towards a Risk-Based Security Requirements Engineering Framework. In 11th International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'05), 2005.

[27] P. Giorgini, F. Massacci, J. Mylopoulos, and N. Zannone. “Modeling security requirements through ownership, permission and delegation”. In Proceedings of the 13th IEEE International Conference on Requirements Engineering (RE'05), pp. 167-176, 2005.

[28] G. Elahi. *Security Requirements Engineering: State of the Art and Practice and Challenges*. 2009.

[29] S.E. Page. *Understanding Complexity. The Great Courses*. Chantilly, VA, USA: The Teaching Company, 2009.

[30] D.A. Fisher. *An Emergent Perspective on Interoperation in Systems of Systems*, Technical Report, CMU/SEI-2006-TR-003, ESC-TR-2006-003. March 2006.

[31] T-AREA-SoS, Trans-Atlantic Research and Education Agenda on Systems of Systems. [Online]. Available: https://www.tareasos.eu/. [Accessed 6 June 2018].

[32] Road2SoS. [Online]. Available: http://www.road2sos-project.eu/cms/front_content.php. [Accessed: 7 January 2018].

[33] COMPASS. [Online]. Available: http://www.compass-research.eu/. [Accessed: 7 January 2018].

[34] DANSE. [Online]. Available: http://www.danse-ip.eu/home/. [Accessed: 7 January 2018].

[35] J. Dahmann, “System of Systems Pain Points”, *INCOSE International Symposium*, vol. 24, no. 1, pp. 108-121, 2014.

[36] C. Ncube, S.L. Lim, and H. Dogan, “Identifying Top Challenges for International Research on Requirements Engineering for Systems of Systems Engineering”, In Proceedings of the *21st IEEE International Requirements Engineering Conference*, pp. 342-344, 2013.

[37] C. Ncube, “On the engineering of systems of systems: key challenges for the requirements engineering community”, Workshop on IEEE Requirements Engineering for Systems,

Services and Systems-of-Systems (RESS), Trento, Italy, August 30, 2011.

[38] M. Hapgood, *Lloyd’s 360° Risk Insight Space Weather: its impact on Earth and implications for business*. 2011.

[39] W. Hurst and A. MacDermott, “Evaluating the effects of cascading failures in a network of critical infrastructures”, International Journal of System of Systems Engineering, vol. 6, no. 3, pp. 221-236, 2015.

[40] The UK Government Office for Science. *Distributed Ledger Technology: beyond block chain*. December 2015.

[41] L. Lamport, and P.M. Melliar-Smith. Byzantine Clock Synchronization, Proceedings of the third annual ACM symposium on Principles of Distributed Computing, 1984.

[42] P. Eeles. Capturing architectural requirements. [Online]. Available:http://www.ibm.com/developerworks/rational/library/4706.html. [Accessed 25 January 2018].

[43] M. Jamshidi, “System-of-Systems Engineering – A Definition”, *IEEE SMC*, 2005.

[44] The Open Group, TOGAF™ - The Open Group Architecture Framework.[Online].Available:http://pubs.opengroup.org/architecture/togaf8-doc/arch/. [Accessed 25 January 2018].

[45] B. Nuseibeh, “Weaving together requirements and architectures”, *Computer*, vol. 34, no. 3, pp. 115-119, 2001.

[46] W. Engelsman, H. Jonkers, H.M. Franken, and M.E. Iacob, “Architecture-driven requirements engineering”. In *Working Conference on Practice-Driven Research on Enterprise Transformation*. Springer, Berlin, Heidelberg pp. 134-154, 2009.

[47] S. Robertson and J. Robertson, *Mastering the Requirements Engineering Process*, Addison-Wesley Longman, 1999.

[48] G.A. Lewis, E. Morris, P. Place, S. Simanta, and D.B. Smith, “Requirement Engineering for Systems of Systems”, *3rd Annual IEEE Systems Conference*, Vancouver, Canada, 2009.

[49] J. Holt, S. Perry, R. Payne, J. Bryans, S. Hallerstede, and F.O. Hansen, “A Model-Based Approach for Requirements Engineering for Systems of Systems”, *IEEE Systems Journal*, vol. 9, no. 1, pp. 252-262, 2015.

[50] M. Jarke, J. Bubenko, C. Rolland, A. Sutcliffe, and Y. Vassilou, “Theories underlying requirements engineering: An overview of NATURE at genesis”, In Proceedings of the *IEEE International Symposium on Requirements Engineering*, pp. 19-31, 1993.

[51] Unified Modeling Language (UML), ver. 2.3. Object Management Group, 2010.

[52] Systems Modeling Language (SysML), ver. 1.2. Object Management Group, 2010.

[53] S. Greenspan, *Requirements Modeling: A Knowledge Representation Approach to Requirements Definition*, Ph.D. thesis, Department of Computer Science, University of Toronto, 1984.

[54] B. Berenbach and M. Gall, “Toward a Unified Model for Requirements Engineering”, In Proceedings of the *2006 IEEE International Conference on Global Software Engineering (ICGSE'06)*, pp. 237-238, 2006.

[55] M. Glinz and D. Wüest, *A Vision of an Ultralightweight Requirements Modeling Language*. Zürich, Switzerland: University of Zurich, Department of Informatics (IFI), 2010.

[56] J. Cleland-Huang, C. Chang, and M. Christensen, “Event-Based Traceability for Managing Evolutionary Change”, *IEEE Transactions on Software Engineering*, vol. 29 no. 9, pp. 2003

[57] A. Kannenberg and H. Saiedian, “Why Software Requirements Traceability Remains a Challenge”, *CrossTalk The Journal of Defense Software Engineering*, vol. 22 no. 5, pp. 14-19, 2009.

[58] L. Northrop, P. Feiler, R. Gabriel, J. Goodenough, R. Linger, T. Longstaff, R. Kazman, M. Klein, D. Schmidt, K. Sullivan, and K. Wallnau, *Ultra-large-scale Systems: The Software Challenge of the Future*. SEI, Carnegie Mellon. 2006.

[59] Sabine Hafner-Zimmermann and Michael J. de C. Henshaw, The future of Trans-Atlantic Collaboration in Modelling and Simulation of Cyber-Physical Systems, A Strategic Research Agenda for Collaboration, 2017, Steinbeis-Edition, Stuttgart, http://www.tams4cps.eu/

[60] Kande Akshay, Integration of Model-Based Systems Engineering and Virtual Engineering Tools for Detailed Design" (2011). Masters Theses. 5155. http://scholarsmine.mst.edu/masters_theses/5155

[61] A World in Motion, INCOSE Systems Engineering Vision 2025, International Council on Systems Engineering, 2014

[62] Golnaz Elahi and Eric Yu, A Semi-Automated Tool for Requirements Trade-off Analysis (2011), Pre-Proceedings of the 23rd International Conference on Advanced Information Systems Engineering Forum, London, UK, June 20 - 24, 2011

[63] Günther Ruhe, Armin Eberlein, and Dietmar Pfahl, Trade-off Analysis for Requirements Selection, Int. J. Soft. Eng. Knowl. Eng. 13, 345 (2003).

[64] Jonathan Lee and Jong-Yih Kuo, New Approach to Requirements Trade-Off Analysis for Complex Systems, IEEE Transactions on Knowledge and Data Engineering, Vol. 10, No 4, 1998

[65] Golnaz Elahi and Eric Yu, Requirements Trade-offs Analysis in the Absence of Quantitative Measures: A Heuristic Method, 26TH ACM Symposium on Applied Computing (SAC’11), March 21-25, 2011, TaiChung, Taiwan

[66] B.H. Cheng, and J.M. Atlee, “Research directions in requirements engineering”, In *2007 Future of Software Engineering (FOSE '07)*, pp. 285-303, 2007.

[67] X. Franch, A. Maté, J.C. Trujillo, and C. Cares, “On the joint use of i* with other modelling frameworks: a vision paper”, In Proceedings of the IEEE 19th International Requirements Engineering Conference, pp. 133-142, 2011.

[68] A. Milne, and N. Maiden, “Power and Politics in Requirements Engineering: A Proposed Research Agenda”, In Proceedings of the *IEEE 19th International Requirements Engineering Conference*, pp. 187-196, 2011.