A research agenda for systems of systems architecting

Ricardo Valerdi,* Elliot Axelband, Thomas Baehren, Barry Boehm, Dave Dorenbos, Scott Jackson, Azad Madni, Gerald Nadler, Paul Robitaille and Stan Settles

Engineering Systems Division,
Massachusetts Institute of Technology,
77 Ave 41-205,
Cambridge, MA 02139, USA
Fax: +1-617-258-7845
E-mail: rvalerdi@mit.edu
E-mail: elliot_axelband@rand.org
E-mail: thomas.baehren@de.bosch.com
E-mail: boehm@usc.edu
E-mail: dave.dorenbos@motorola.com
E-mail: jackessone@cox.net
E-mail: amadni@IntelSysTech.com
E-mail: nadler@usc.edu
E-mail: paul.robitaille@lmco.com
E-mail: settles@usc.edu
*Corresponding author

Abstract: This paper, documents the activity of a workshop on defining a research agenda for Systems of Systems SoS; Architecting, which was held at USC in October 2006. After two days of invited talks on critical success factors for SoS engineering, the authors of this paper convened for one day to brainstorm topics for the purpose of shaping the near-term research agenda of the newly convened USC Center for Systems and Software Engineering (CSSE). The output from the workshop is a list of ten high-impact items with corresponding research challenges in the context of SoS Architecting. Each item includes a description of the research challenges, its link to contemporary academic or industrial problems and reasons for advocacy of that area. The items were assessed in terms of value and difficulty to determine a prioritisation both for the CSSE's future research agenda and for others in the field.

Keywords: systems of systems architecting; resilience; complexity; net-centric; Model-Driven Architecting; MDA.

Reference to this paper should be made as follows: Valerdi, R., Axelband, E., Baehren, T., Boehm, B., Dorenbos, D., Jackson, S., Madni, A., Nadler, G., Robitaille, P. and Settles, S. (2008) 'A research agenda for systems of systems architecting', *Int. J. System of Systems Engineering*, Vol. 1, Nos. 1/2, pp.171–188.

Biographical notes: Ricardo Valerdi is a Research Associate in the Engineering Systems Division at MIT and a Member of the Board of Directors of INCOSE.

Elliot Axelband is a Senior Engineer at the RAND Corporation, an Emeritus Professor, Associate Dean and the Former Director of the System Architecting and Engineering Programme at USC. He is a retired Hughes Aircraft VP and a Fellow of both INCOSE and IEEE.

Thomas Baehren is the Director of Corporate Sector Research and Advanced Engineering Systems for Robert Bosch GmbH.

Barry Boehm is the TRW Professor of Software Engineering, the Director of the CSSE at USC, a Member of the National Academy of Engineering, and an INCOSE Fellow.

Dave Dorenbos has worked at Motorola, Gould and Rockwell International in a variety of development and research roles; most recently as the Director of Software and System Engineering Research at Motorola. He is also a Dan Noble Fellow at Motorola.

Scott Jackson is an Adjunct Associate Professor in the Systems Architecting and Engineering graduate programme at USC and an INCOSE Fellow.

Azad Madni is the CEO of Intelligent Systems Technology and a Fellow of IEEE.

Gerald Nadler is IBM Chair Emeritus in Engineering Management and Professor Emeritus of Industrial and Systems Engineering at USC. He is a Member of the National Academy of Engineering and the past president of IIE.

Paul Robitaille is a Systems Fellow at Lockheed Martin and the President of INCOSE.

Stan Settles is the IBM Chair in Engineering Management, Director of the Systems Architecting and Engineering Program and Co-Director of the CSSE at USC. He is the past president and Fellow of IIE, a Fellow of INFORMS and a Member of the National Academy of Engineering.

1 Background

The University of Southern California hosted a convocation celebrating the creation of its new Center for Systems and Software Engineering (CSSE) from 10/23 to10/26, 2006. CSSE has been formed from the prior Center for Software Engineering (CSE) and the prior System Architecting (SA) and Engineering Research Programmes at USC. Their Directors – Barry Boehm and Stan Settles, are the Co-Directors of CSSE and ultimately responsible for its research agenda. The convocation agenda, events, speakers, programmes, presentations and the results of working group sessions, etc. can be found at: <http://csse.usc.edu/events/2006/CSSE_Convocation/pages/home.html>.

There were many notables who presented at the convocation, to include, but by no means is this a complete listing: The President of the National Academy of Engineering,

The President of USC, The Dean of the USC Viterbi School of Engineering, The former DoD CIO and NII Director, The President of INCOSE and the DoD Director for Systems and Software, as well as VPs and other prominent players from commercial and DoD related industry, academe and government.

CSSE is a collaborative entity. Its interests span USC beyond engineering, and its research agenda is intended – and has functioned in its predecessor organisations – to include partners from a broad range of university, industry and government affiliates. Current activities, beyond the scope of this paper, illustrate this.

This paper documents the activity of one of the convocation workshops that was formed to provide a research agenda for Systems of Systems (SoS) Architecting. The authors of this paper were the participants in this workshop. Through a series of lively discussions and debates, culminating in an evaluation and ranking of research candidates, followed by presentation to and discussions with a larger group, a proposed CSSE research agenda was formed, which is provided in this paper.

The remainder of this paper consists of the elements of the research agenda, an evaluation matrix used to assess the value and difficulty of each item and a summary of next steps.

2 Research agenda

A group of 13 participants representing commercial organisations (Bosch, Intelligent Systems Technology and Motorola), defence companies (Lockheed Martin, Northrop Grumman and Boeing) and academia (USC and MIT) convened for a day-long workshop aimed at defining a research agenda for SoS Architecting. This brainstorming session was focused on identifying a set of critical success factors with the following characteristics: degree of improvement over the current body of knowledge in better understanding and practicing SoS architecting; and added value to at least one SoS stakeholder. The items are listed here in the order they were offered to show the evolution of the group's thinking and learning.

1 Resilience

2 Illustration of success

3 System versus SoS attributes

4 Model-driven architecting

5 Multiple SoS architectural Views

6 Human limits to handling complexity

7 Net-centric vulnerability

8 Evolution

9 Guided emergence

10 No single owner SoS

The participants attended two days of presentations where they were exposed to diverse viewpoints of SoS from diverse environments (e.g. space, automotive and telecommunications) and specialties (e.g. software, hardware and human factors). Participants entered the workshop with their own working definitions of a SoS and SoS architecting based on their background and priorities. Ultimately, the brainstorming exercise was intended to identify ideas and refine them to fit the group's charter; to help shape the future of CSSE and other interested research organisations.

2.1 Resilience

This area is the attribute of a system, in this case a SoS that makes it less likely to experience failure and more likely to recover from a major disruption. Challenger, Columbia, Chernobyl and Bhopal are examples of such failures. Resilience engineering considers the process, disciplines, infrastructure and cultural attributes that need to be in place to make failure less likely and recovery more likely. Human factors are a major contributor over the entire life: design, test, production, maintenance, operation, etc. Resilience goes beyond design focused attributes, such as reliability. (Hollnagel et al., 2006) provide comprehensive analyses of resilience and a description of the concepts and precepts pertaining to it. The recommended research pertaining to SoS resilience has four main thrusts:

- Determine the architectural features of the end product that will make it less vulnerable to catastrophic failure in a SoS environment and more adaptable to recovery. SoS, by definition, are federations of systems. The interaction among elements of individual systems may experience emergent behaviour and affect the resilience of the SoS. For this reason, the interactions among these systems need to be carefully architected. According to Woods (2006), the essential characteristic of such a system is adaptability. The architectural methodologies to achieve this adaptability are a challenge of SoS research.

- Determine the architectural aspects of the supporting infrastructure, to include the developer, customer, suppliers, operators and maintainers, to make the resulting end product less vulnerable to catastrophic failure in a SoS environment. Speakers at the CSSE Convocation pointed to the challenges created by the interaction among the many different organisations associated with a SoS. Figure 1 is an example of the use of a Department of Defence Architectural Framework (DoDAF) Operational View (OV-2) to illustrate a resilience framework.

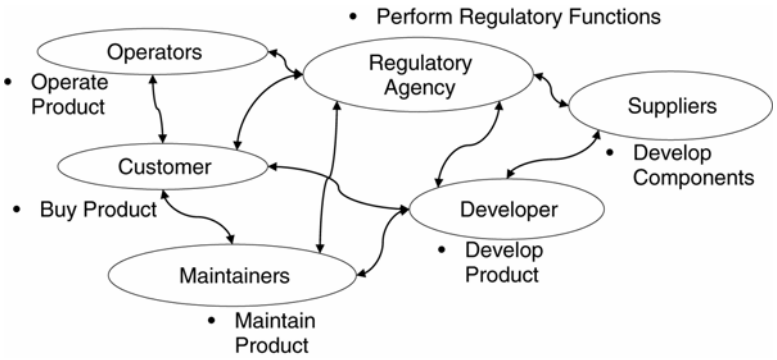
- Identify methodologies to create a culture that will enable system resilience in a SoS environment. Many references, for example, (Vaughn, 1996) document ways in which some cultural paradigms are detrimental to system resilience. Few references provide approaches to cultural change to improve resilience. (Jackson et al., 2006) provide a list of potential methods.

- Develop statistical methods of showing when a system is vulnerable to catastrophic failure based on defects at various phases of the life cycle including architecting, development and operation. Many references, for example, Reason (1997), note that most catastrophic accidents are preceded by minor defects or near misses. The logical conclusion is that a statistical correlation can be made between the minor defects and the catastrophes. If this correlation can be made, then the following benefits may result: firstly, SoS prone to disaster can be identified in advance.

Secondly, minor defects in both processes and operation can be used to manage the programme to maximise resilience. Wright and Van der Schaaf (2002) have shown that such a correlation does exist.

Reason for advocacy: SoS, because of their complexity, the dynamic state of their design characteristics and their emergent behaviour are particularly vulnerable to catastrophic failure. For this reason, the four research architecting-focused topics listed above promise to yield substantial benefits to SoS.

Figure 1 Operational view of the system resilience infrastructure (OV-2)



2.2 Illustration of success

Eberhardt Rechtin, the founder of the SA movement, asked the following question in a 24 March 2005 e-mail to Prof. Nadler:

> "When I first came to USC I came with a question from fellow CEO's "WHY is it that, although we have created and recreated the finest engineers, managers and scientists in the world, and although we have created some of the finest world-class, memorable projects (and some of the greatest disasters as well) we can't tell ahead of time whether they will be glorious successes OR terrible failures?"

Robert W. Lucky, retired vice president of Telcordia Technology and IEEE Fellow, stated in his article 'Unsytematic Engineering' in the September 2006 *IEEE Spectrum* (p. 84) that

> "….the way we go about engineering large systems [:] Divide and conquer is the usual approach….Any possibility of a holistic approach is foregone from the very start."

SoS architecting is an acknowledged complex process and holistic thinking effort. Developing a teachable and implementable model of *how* SoS architecting ought to be handled by an individual or group would be of significant value in responding to the issues raised by Rechtin and Lucky as well as many others in the SA business. Interestingly, the heuristics developed by Rechtin in his pioneering book on SA were insufficient for him to answer his own question.

This research initiative adds a missing critical dimension to the literature on studies of large-scale engineering and systems projects. Almost all of them provide retrospective analyses of what went wrong in large projects. For example, Henry Petroski's

'To Engineer Is Human: The Role of Failure in Successful Design' reports on disasters of large-scale engineering systems. Paul Nutt's *Why Decisions Fail* reports the results of around 400 major cases in all types of organisations.

Previous research (e.g. Peterson, Sakman, Murtha, Friedman and Bennett et al) used various methodologies to determine *how* leading creators of 'regular' systems (such as engineers, architects, information systems designers and product developers) thought about and proceeded on projects. This research led to a synthesis of a holistic approach to these types of planning and design projects that have resulted in significantly better solutions than developed with the 'divide and conquer' approach.

We propose that these research methodologies can be adapted and expanded to study *successful* SoS architecting efforts to provide a basis for a similar synthesis. We hypothesise that the outcomes of this research will be more beneficial to teaching and doing SoS architecting than using only studies of failures. The latter studies report only on what went wrong in the belief that future SoS architects will thus avoid similar pitfalls and do not provide an overall mode of reasoning.

2.3 System versus SoS attributes

Traditional systems engineering as defined in the definitive text *Systems Engineering and Analysis* (Blanchard and Fabrycky, 1998) considers three main ilities: reliability, maintainability and supportability. These are adequate for situations where systems are product oriented and architects have control over the constraint space. In the case of SoS, however, additional ilities need to be considered because the unit of analysis is more complex and the solution space is larger. In addition, technical attributes need to be considered in concert with socio-technical issues.

The 1973 TRW *Characteristics of Software Quality* study for the National Bureau of Standards identified and analysed seven primary software quality factors in the centre of a three-level hierarchy: reliability, efficiency, usability, testability, understandability, modifiability and portability (Boehm et al., 1973). The Engineering Systems Division at MIT has defined a set of ilities that should be considered throughout the life cycle of engineering systems (Allen et al., 2002) a subset of which are adequate for this SoS discussion. The ilities in general are not limited to words whose suffix is 'ilities'. The term is broadly used to include requirements of systems that are not necessarily part of the fundamental set of functions or constraints. Examples of ilities include:

- *Adaptability* : the ability of a system to change internally and undergo self-modification.
- *Flexibility* : the property of a system that is capable of undergoing changes based on the external environment with relative ease.
- *Agility* : ability of a system to be both flexible and undergo change rapidly.
- *Scalability*: the ability of a system to maintain its performance and function, and retain all its desired properties when its scale is increased greatly without a corresponding increase in the system's complexity.
- *Modularity*: the degree to which the components of a system can be designed, made, operated and changed independently of each other.
- *Sustainability*: maintaining economic growth and viability while meeting concerns for environmental protection, quality of life and social equity.

The implications of these ilities on the architecting of SoS are threefold. Firstly, the system architect must identify and manage the ilities influenced by the success critical stakeholders throughout the life cycle. This becomes a more difficult task when the unit of analysis is at a higher level of abstraction as in the case of the US Army's Future Combat System where the number of stakeholders is in the 100s.

> Research challenge #1: How can an SoS architect identify and manage the broad range of ilities inherent in the SoS (explicit) and introduced by the operational environment (implicit)?

Secondly, the simulated architecture of the SoS must include these constraints when making tradeoffs between ilities. Rather than addressing a single ility at a time, the SoS architecting process must consider the dynamics and interactions of their collective existence. For example, replicating and distributing data may enhance resilience, but may reduce security; protection layers may enhance security but reduce performance and tightly coupled architectures may enhance performance but degrade resilience.

> Research challenge #2: How can an SoS architecture be modelled to include the ability to perform tradeoffs between ilities?

Finally, the SoS ilities and the constraints they introduce must be quantified and validated in some way. This involves the operationalisation of each ility's definition and subsequently effective ways of testing them in isolation and as a cluster. It is often forgotten that ility metrics are only meaningful with respect to an operational context, such as response time or mean time between failures.

> Research challenge #3: How can SoS ilities be measured and tested?

Reasons for advocacy: By developing methods to identify, manage and measure SoS ilities, system architects will be able to predict the behaviour and ultimately the success of an SoS.

2.4 Model driven architecting

Model-driven approaches are becoming increasingly attractive as software systems continue to grow in size, complexity and at the same time demand greater attention to the ilities. Since the primary driver of increasing complexity in modern and future systems is the triad of computer science, software and communications technologies, it is fitting that we also look to these technologies to strengthen systems engineering's ability to perform effective MDA in order to manage this complexity. Simple and unguided extrapolation to MDA practice is unlikely to provide the needed capabilities: systems are upgraded at a pace determined by the marketing plans of the software developers; they are poorly integrated across functions which must work intimately and; for the most part, they merely mechanise human processes – at a far faster pace – rather than improve upon them.

Model-driven approaches including MDA™ as well as Model-Driven Development (MDD™) techniques hold the promise to vastly improve the quality of future software systems due to their formal underpinnings and support for roundtrip engineering as well as automated code generations and testing. The Object Management Group (OMG), a leading Consortium of commercial tool vendors including IBM, Borland and TeleLogic

is committed to advancing a set of standards that collectively underpin model-driven methodologies and technologies (OMG, 2001). Many commercial vendors are leveraging these standards to implement their respective brands of MDA and MDD tools and technologies. The OMG vision of MDA is to define system functionality as a Platform-Independent Model (PIM) using a domain-specific language. A PIM is then transformed into a Platform-Specific Models (PSMs) intended for a particular runtime environment (e.g. J2EE or .NET) and implementation using a domain-specific or general-purpose language such as Java, C++ or C#.

Model transformation is usually accomplished using automated means, a potential productivity multiplier. Common to all model-driven approaches is a set of standards for structuring, representing, visualising, expressing and storing software systems specifications as models (Madni et al., 2006). While few disagree about the merits of developing a formal model of software specifications, realising the full potential of MDA remains a challenge especially for large-scale software systems (Bézivin et al., 2003). The Gartner Group has identified MDA technology as 'being on the rise' (Gartner, 2006). Many development tool vendors continue to press on with expanding their tools support for MDA and MDD. These tools cover model creation, analysis, transformation, composition, test, simulation, metadata management and reverse engineering. Specifically, the research arm of IBM is continuing to work on overcoming obstacles and gaps in MDA approaches. New advances in model-driven approaches continue to be reported in a variety of professional and research conferences such as ECMDA, OOPSLA and ECOOP. In all, model-driven approaches to software systems architecture modelling and development are expected to bear significant fruit in the next several years.

Against the foregoing backdrop, there is a need for:

1 Analytical models for estimating cost, schedule, quality, productivity and other value attributes associated with applying model-driven approaches to developing large-scale software-intensive systems. Such a capability could be built on top of Model-Based System Architecting and Software Engineering (MBASE) guidelines (MBASE, 2006). It might be the case that existing cost, size, quality estimation approaches may have to be redefined, revised or extended in light of the fact that, with model-driven approaches, code is generated automatically through the application of technology-specific transformation. Specifically, research is needed to determine the parameters that allow us to predict cost, productivity and other value attributes and metrics associated with MDA and MDD projects.

2 Methods to determine and validate whether existing models can be modified for this purpose or whether a new model and a different cost framework are required. The expected benefits of model-driven approaches are increased productivity, greater traceability and lower development cost and risk.

3 Multidimensional Mathematical Model-Manager methods and tools, employing graph theory–and its offshoot, constraint theory–to determine model consistency and computational 'allowability' within models containing tens of thousands of variables. This capability should be applied to enhance the precision and mathematical foundation for modelling languages such as SysML.

4 Evolutionary computation and generic algorithms to search the vast trade space for satisfying designs.

5 Quantitative risk management, based on decision theory, to converge on designs with the balance of cost, performance and risk preferred by the stakeholders.

6 Value and preference models to translate the diverse requirements of the stakeholders as well as their risk assessments into acceptance test standards that the model can verify.

Very little research has so far been invested in the development of such models. Even partial success with respect to the above vectors promises substantial improvement in MBSE and brings systems engineering closer to the long-term goal of a 'Unified Theory of Systems Engineering'.

2.5 Multiple SoS architectural views

It is well-known that having multiple views of a complex system can help people reason about, decide and manage system issues. Doctors and their patients are helped by having views of the human body's skeletal, muscular, digestive, blood flow and nerve subsystems. Building or aircraft developers and their stakeholders are helped by having views of the building or aircraft's spatial, structural, mechanical, hydraulic and electrical subsystems. Software developers and their stakeholders are helped by having views of the software's data flows, state transitions, class hierarchies, physical deployment and usage views.

For individual systems, reasoning about the consistency, compatibility and feasibility of these views is difficult enough. However, particularly for network-centric SoS involving many large, closely coupled, separately evolving, multimission systems and their stakeholders as components, these difficulties become much formidable. The larger and more complex the SoS, the more useful and important are these multiview capabilities, but also the more difficult it becomes to represent and apply the views.

When SoS developers and their stakeholders reach onto the shelf for architectural representations, tools, processes and methods to deal with these multiview issues, they find that current technology has significant shortfalls. Particularly challenging and high-leverage research topics for improving these capabilities are:

- *Scalability*: for example, a network-centric SOS can have an enormous number of states and state transitions. Some of the transitions may be catastrophic, but particularly for discrete digital systems, there are no strongly scalable techniques for recognising and avoiding them. Techniques such as suppressing detail of 'distant' systems and their states can reduce the scale, but with uncertain effects on thoroughness of undesired-state diagnosis and avoidance.

- *The ilities in general*: specialised views for reasoning about SoS performance, reliability, usability and other ilities can be quite helpful. But as discussed in Topic 3, scalable techniques for reasoning about SoS ilities and their interactions are much needed. And the more views one has, the more complex it becomes to reconcile their representations, assumptions and interactions.

- *View consistency assurance*: some low-hanging-fruit research has been done in areas such as traceability across problem views and solution views, and architecture style compatibility analysis. But more general capabilities for scalable SoS view consistency assurance in such areas as constraint satisfaction, concurrency effects, assumption compatibility and composability are much needed.

- *View update propagation*: a network-centric SoS will be continually evolving, requiring views of it to be easy to update, including propagation of side effects across different views and systems. Incremental analysis tools that do not require re-doing the full cross-view analysis for small changes will be increasingly valuable.
- *Unviewables*: a network-centric SoS will have an increasing number of increasingly complex, proprietary, Commercial-Off-The-Shelf (COTS) products whose internals are not viewable. Better interface and assumption definition capabilities for such products and scalable service-oriented architectures will be increasingly important.

2.6 Human limits to handling complexity

The sheer complexity of SoS such as the Air and Space Operations Center (AOC) and the need to adapt to changing environmental conditions of SoSs pose serious challenges to individual and collaborative (i.e. team) decision making. Failed decisions are invariably a consequence of: rush to judgement; misuse of resources and repeated use of failure-prone tactics (Nutt, 2002).

Responding to unexpected challenges with flexibility and reducing the disruptive effects of change requires using tools such as sensemaking, stress reduction, decision migration and labelling which allow individuals and teams to be 'mindful' – that is, alert, resilient and flexible (Weick and Sutcliffe, 2001). They emphasise learning to 'notice the unexpected in the making and halt its development'. In other words, they show how to detect aberrant conditions while they are 'new, small and insignificant' before they become highly consequential.

Finally, it is important to note that individuals within teams behave quite differently than when acting alone. Considerations such as risk taking propensity, trust and socio-cultural factors all come into play (Madni, 2006). Are the ways people collaborate and interact (in the SoS Architecting domain) so different from the way they do in the System Engineering world?

As a result, we suggest it is time to go beyond almost all the discussion about SoS factors that differentiate them from 'regular' systems engineering efforts – for example, they are complex, dynamic, changing and highly interdependent, and have several purposes and a variety of stakeholders to support. Mentioned mostly in passing are the characteristics of the people related to the SoS – developers, users, implementers, operators, information technologists and maintainers. Yet virtually every reference to the potential effectiveness of the SoS mentions that 'people are the problem'.

If long term usefulness and success of future SoS are to be significantly improved, much knowledge must become available about the abilities of people in their various roles, especially related to their reactions to and usefulness of abilities in the SoS, or on the scalability of human-system analysis techniques to SoS, such as cognitive task analysis (Cummings, 2006). Some useful work in this direction has been done, such as in Booher (2003), but a great deal more needs to be done.

A candidate research initiative seeks to develop a useable database about human attributes that are found to be essential in handling complexity and change (NAVY, 1998). SoS architects can use the database to more fully specify the human behaviour and skill requirements for and elements of the SoS, especially on the changes needed to adopt and operate the expected continually evolving SoS. Other candidate

research issues include SoS human factors test beds and human factors engineering capabilities for collaborative multimission systems.

The hypothesis of this research is that the identified human attributes will enable significantly better education of future systems architects and improved SoS architecting in all fields. This would allow SoS architects to define a typology of human attributes in SoS architecting, operation and maintenance; and design appropriate methods in the form of decision aids and mechanisms to build on them. A companion research goal is to identify the typology of unexpected SoS behaviours through, for example, SoS modelling and simulation, and develop aids building on the attributes for humans to detect their occurrence and take appropriate containment action. Finally, a typology of attributes and patterns that characterise successful ad hoc teams needs to be identified and used as a basis to calibrate and improve collaborative SoS teams. The domain for such an investigation could be a reasonably structured SoS such as the AOC or an emergency management SoS (Jones et al., 2004).

2.7 Net-centric vulnerability

The world is becoming flat (Friedman, 2005), tied together by trusted networks that enable distance world-wide collaboration and near instantaneous access to data and the means to evaluate data in context, and act decisively, anywhere, any time, with precision and accuracy. This vision has become a strategy that permeates both commercial and the defence enterprise, where it is called Net-Centric Warfare. Information technology (IT) intensive SoS exist and are being developed incorporating this premise.

But a flat world is not a friendly world, and trust cannot be assured. Networks undergo information attacks that manifest themselves in various forms such as viruses and worms that plague the internet, and reduce its efficiency in various ways, from the time required to debug and repopulate infected computers, to the expense of network defence systems that by their very nature reduce network efficiency. And these are the simple problems. Network intrusion has caused the loss of identity, the theft of massive 'private' databases, and the compromise of 'secure' data and systems. Malicious software injected into IT systems can be difficult to detect, let alone defend against, is rapidly evolving and has become a tool used by some governments to prey upon the commercial and defence activities of others.

This situation is well recognised and is a direct consequence of the IT properties of networks as currently configured, and the means by which they are enabled by software. Clearly, the current IT paradigm – or its implementation – must change, if that flat world is to avoid a chasm filled future. While these problems are recognised and there is research being undertaken, there is more to be done and the potential contribution of SoS Architecting must be developed and understood. We are aware of no research taking this full perspective at this time.

2.8 Evolution

One of the basic characteristics of a SoS is that it is evolutionary, that is, it develops over time and not over a fixed development period. Sometimes it may evolve within a predicted evolution envelope but in most cases the SoS may develop emergent

characteristics, that is, characteristics that were not planned for. Emergence can be a negative property of a SoS especially if it is unpredictable. When such a property appears, it must be managed since it can rarely be avoided.

Since SoS are rarely planned, controlling these emergent characteristics is usually not feasible. The thought is that steering the SoS by applying constraints and incentives is a more viable approach. The question becomes: what mechanisms can be applied to apply the constraints and incentives? One way that is presently being used to facilitate evolution is open architectures and open sources. Research may identify others in the future. Some of the suggested approaches to this problem are as follows:

- the SoS must have adaptive features, that is, they must be able to adapt dynamically to emergent situations or to changing desired capabilities
- the architecture must have 'buffers' to absorb the deviation from the evolution envelope of the SoS.

It is realised that the above solutions may not fit within the reductionist methodologies of traditional systems engineering. However, such new methodologies will become part of an expanded systems engineering of the Systems Engineering Vision (INCOSE, 2006).

An important aspect of any evolutionary SoS is the contractual aspects. It was noted that most procurement at the present is based on the assumption that a developer will provide a product based on documented set of needs that are converted into a specified set of requirements. If a system of system is constantly evolving, then the current system of contracting becomes inadequate. Two methods suggested for improving this process are:

- stick with a single team over a large period of time, that is, over several variations in the SoS
- insert agile provisions in the contract, that is, avoid specific requirements and focus on long-term development
- concurrently engage stabilised incremental development teams and agile next-increment replanning and rearchitecting teams, under different contractual structures (Boehm and Lane, 2006).

It was also noted that whatever solution is arrived at, it must be acceptable to the acquisition community within which the terms 'evolutionary' and 'spiral' have not been looked on favourably. In short, the research envisioned for SoS evolution includes:

- research devoted to the development of new systems engineering methodologies to cope with SoS evolution and emergent properties associated with it.
- research associated with the development of new contractual mechanisms to handle SoS evolution.
- research on improved processes better employing the principles underlying evolutionary and spiral acquisition and development: synchronised and stabilised concurrent engineering, risk management, stakeholder satisficing, iterative SoS definition and development.

Reason for advocacy: the above research is essential for dealing with one of the major characteristics of SoS, that is, evolution. Without it, SoS programme failures can be anticipated.

2.9 Guided emergence

Guided emergence, a term coined by Madni (2006), is the ability to steer emergent behaviour in desired directions. Within the context of SoS architecting, guided emergence can be viewed as a strategy to achieve mission objectives, that is, the goals of a SoS. To appreciate the concept of guided emergence, let us examine how a city comes into being. Cities are not imagined and built by individual organisations. Rather, they come into being as a result of the decisions and actions of many individuals acting locally over extended periods of time. The key factors that go into how a city comes together are given in Table .1

In this example, if one were to replace the term 'building' with 'system' and 'city' with 'system-of-system', one will find that the parallels between cities and SoSs hold rather well. It therefore follows that using the metaphor of a city and how it comes into being should shed light on SoS architecting. Specifically, research is needed into designing mechanisms that can guide emergent SoS behaviours in directions that satisfy SoS objectives. The first research area is the *design of mechanisms* (CMU/SEI, 2006) that can achieve globally optimal behaviour when participating agents continue to act in their own self-interests regardless of the interests of other participating agents.

The second area of research is motivated by the fact that SoS evolution can be expected to be an ongoing, dynamic phenomenon. Research is needed into the rules needed to enable SoS evolution and how to build these rules into the SoS and its constituent systems, processes and tools so that the SoS is able to adapt to dynamically changing environment without requiring constant human intervention.

Table 1 The emergence of a city

• Cities are not conceived or built by individual organisations and are not merely a collection of buildings.
• The form of a city is not specified in advance.
• A city emerges and changes over time through the loosely coordinated and regulated action of individuals to satisfy the needs of its citizenry.
• The ingredients that cause a city to operate and grow indefinitely are its infrastructure beyond that present in individual buildings and mechanisms (e.g. policies and regulations) that regulate local action to achieve and maintain operational coherence in the absence of centralised control.
• Various agents (e.g. people, communities, organisations) build certain parts of the cities to satisfy their respective objectives.
• Cities grow and flourish based on societal and economic stimulus or they falter and fall into decay when such stimulus is absent.
• While some aspects of a city are designed and constructed in a local context, most aspects are a result of global policies (e.g. zoning ordinances, business incentives).

2.10 No single owner SoS

Almost all SA and SoS Architecting literature address 'single owner' conditions – an automobile or airplane platform, a combat system. However, there are many SoSs that do

not fit this category yet are in desperate need for overall architecting. Some successes without an overarching single-owner, such as the world wide web, wikipedia and Linux operating system, have emerged with a conscious open architecting approach.

Yet, there are many SoSs where it would be highly desirable to consider an identifiable architecting approach, such as automotive transportation, education, healthcare, mortgage and finance, philanthropic organisations, non-government organisations (CARE, Red Cross, United Way, etc.), and government (particularly city and county).

The hypothesis of this research is that a study of successful no-single-owner SoSs would identify characteristics of the architecting process used, the individual and team attributes, the 'culture' that was developed, and related factors that could be used in future SoS architecting situations. Further topics of interest are business models/drivers, safety and security aspects, minimum administration needed (e.g. Domain name ownership in the world wide web) and others that have impact on the architecture development of such SoS.

As an example, the County of Los Angeles (10 million population, $20 billion budget, 90,000 employees, 38 departments) was able to architect a county-wide strategic plan where a five member Board of Supervisors (BoS) has legislative, executive and pseudo-judicial authority. The Chief Administrative Officer (CAO) has no hiring/firing authority – only the BoS can do this. Yet, the many departmental silo heads collaborated to develop a continuing strategic planning approach that has resulted in significant improvements in County services and productivity (see annual reports of the Los Angeles County Quality and Productivity Commission).

Similarly, several NGOs have architected effective SoSs that could be subjected to the investigations proposed in this research. Methodologies that have been successful in studying single owner SoS (data collection and analysis, survey instruments, comparative reviews, synthesis modelling) could be adapted to the large perspectives of SoSs.

3 An evaluation matrix of the proposed research

After the 10 research items were identified, participants were asked to provide a subjective assessment of the difficulty and value of the research. The difficulty of the research was discussed in the framework of three dimensions: intrinsic difficulty, resources and funding. The fact that the systems of systems area is an emerging field presents its own challenges, most of which are captured here.

3.1 Intrinsic difficulty

The intrinsic difficulty of the topic (tractability) and the length of time required to obtain results. Additionally, whether research could be performed by a PhD student or Post Doc.

3.2 Resources

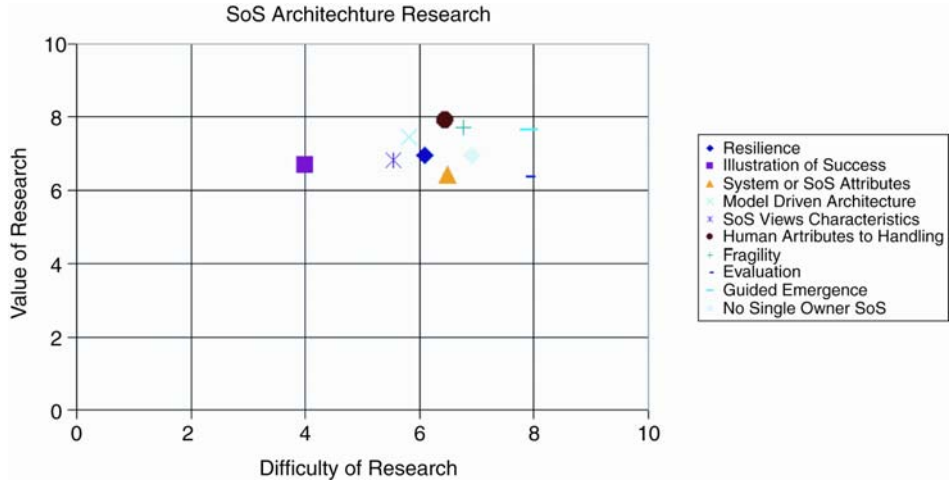
The availability of resources data or documentation needed to perform research (staying away from security-classified data). Necessary equipment/facilities and the degree to which tools are needed to perform the work.

3.3 Funding

The existence and scale of funding required. Each of the 13 experts was asked to consider these aspects of each of the 10 topics and provide an overall rating from 0 to 10. These ratings were later compiled, averaged and plotted in these dimensions as shown in Figure 2. Each quadrant was considered to have certain properties. The first quadrant (top right) included difficult and valuable research. The second quadrant (top left) included low-hanging fruit because of its relatively low difficulty and high value. The third quadrant (bottom left) included research of low value and low difficulty which was not a priority. Finally, the fourth quadrant (bottom right) included research that was considered to be the final resort because of its high difficulty but low value to stakeholders.

Most research topics were clustered on the first quadrant, which successfully reflects the workshop's objectives. This prioritisation exercise should help the CSSE, its Affiliates, and other organisations to define their next steps as they develop research agendas.

Figure 2 Difficulty and value of proposed topics



4 Summary and next steps

This collaborative approach to developing a research agenda proved to be an effective way to articulate priorities and iterate them with experts with diverse backgrounds. As an immediate next step, it is necessary to reconcile the ten items on SoS Architecting with the parallel workshop that took place on the topic of SoS process (Lane and Turner, 2006). A summary of that workshop's recommendations is provided in Appendix A.

In the future, there are additional questions to consider for the SoS Architecting topics in particular. For example, what problem is SoS supposed to solve or suppress? What capabilities must an SoS have to do so? Can thinking about a system by an architect suffice or must the reductionist, prescient design mode give way to autopoiesis? What value would pertain to a ready capability to create SoS as needed?

As the CSSE proceeds on its chosen trajectory of research it must also consider what the measures of effectiveness are for a SoS research programme. This is an important consideration in light of the impact the research can have in academia, industry and government.

Acknowledgements

We would like to acknowledge the participation of other individuals during the workshop on SoS Architecting Research: Jesal Bhuta, Winsor Brown, Ed Colbert, Vu Nguyen, Rod Robertson and Thomas Tran.

References

Allen, T., McGowan, D., Moses, J., Magee, C., Hastings, D., Moavenzadeh, F., Lloyd, S., Nightingale, D., Little, J., Roos, D. and Whitney, D. (2002) 'ESD terms and definitions, version 12', *MIT Engineering Systems Division Working Paper (ESD-WP-2002-1).*

Bézivin, J., Gérard, S., Muller, P-A. and Rioux, L. (2003) '*MDA components: challenges and opportunities'*, Metamodelling for MDA.

Blanchard, B.S. and Fabrycky, W.J. (1998) *Systems Engineering and Analysis*, 3rd edition, Upper Saddle River, NJ: Prentice Hall.

Boehm, B., Brown, J., Kaspar, H., Lipow, M., MacLeod, G. and Merritt, M. (1973) 'Characteristics of software quality', *TRW Report for NBS* (also North Holland, 1978).

Boehm, B. and Lane, J. (2006) '21st century processes for acquiring 21st century software-intensive systems of systems', *CrossTalk*, Vol. 19, No. 5, pp.4–9.

Booher, H. (Ed) (2003) *Handbook of Human Systems Integration*, Wiley.

CMU/SEI. (2006) *Ultra-Large Scale Systems: The Software Challenge of the Future*, CMU Software Engineering Institute.

Cummings, M.L. (2006) 'Can CWA inform the design of networked intelligent systems?' Paper Presented at the Moving Autonomy Forward Conference, Lincoln, UK.

Friedman, T.L. (2005) *The World is Flat,* 1st edition, Farrar, Straus and Giroux.

Gartner Group (2006) *Hype Cycle for Emerging Technologies*.

Hollnagel, E., Woods, D.D. and Leveson, N. (Eds.) (2006) *Resilience Engineering: Concepts and Precepts*, Ashgate Publishing Company.

Jackson, S., Erlick, K. and Gutierrez, J. (2006) 'The science of organizational psychology applied to mission assurance', *Conference on Systems Engineering Research*, Los Angeles, CA.

Jones, R.E.T., McNeese, M.D., Connors, E.S., Jefferson, T. and Hall, D.L. (2004) 'Distributed cognition simulation involving homeland security and defense: the development of NeoCITIES,' *Proceedings of the Human Factors and Economics Society 48th Annual Meeting.* pp. 631–634.

Lane, J. and Turner, R. (2006) *'SoS Process, Acquisition, Management Critical Success Factors Workshop'*, CSSE Convocation Outbrief.

Madni, A.M. (2006) *'Cognitecture™: Cognition-based, Integrative Architecture for Next Generation Intelligent Agents'*, ISTI-FTR-567-08/06.

Madni, A.M., Moini, A. and Madni, C.C. (2006) 'Towards a executable specification-driven toolset for analyzing DoDAF-compliant architectures', *Proceedings of the Ninth World Conference on Integrated Design and Process Technology, (IDPT-2006)*, San Diego, CA.

Madni, A.M. (2006) 'SoS architecting: critical success fFactors', *USC-CSSE Convocation, Executive Workshop Presentation.*

MBASE (2006) Available at http://sunset.usc.edu/csse/research/mbase/.

NAVY (1998) *Recapitalizing the Navy: A Strategy for Managing the Infrastructure*, National Academy Press, Chapter 4 and Appendix D address processes for getting changes adopted.

Nutt, P.C. (2002) *Why Decisions Fail: Avoiding Blunders and Traps That Lead to Debacles*, Berrett –Koehler.

OMG (2001) 'OMG pursues new strategic direction to build on success of past efforts'.

Reason, J.T. (1997) *Managing the Risks of Organizational Accidents*, Ashgate Publishing Company, London.

INCOSE (2006) 'Systems engineering technical vision', Available at http://www.incose.org.

Vaughan, D. (1997) *The Challenger Launch Decision: Risky Technology, Culture, and Deviance at NASA*, University of Chicago Press, Chicago, IL.

Weick, K.E. and Sutcliffe, K.M. (2001) *Managing the Unexpected: Assuring High Performance in an Age of Complexity*, Jossey-Bass.

Woods, D. (2006) 'Essential characteristics of resilience' in Hollnagel. et al,(Eds). *Resilience Engineering*.

Wright, L.and Van der Schaaf, T. (2000) '*Accident versus Near-Miss Causation: A Critical Review of the Literature, an Empirical Test in the UK Railway Domain, and their Implications for Other Sectors', University of Strathclyde.*

Appendix

Adapted from Lane and Turner (2006)

SoS Process Critical Success Factors: Areas for further research

A. Engineering processes that focus on capabilities/outcomes rather than requirements
B. How to fix acquisition process – current process not used right
C. Risk identification templates that can succeed in a SoS context
D. How to do without CRACK[1] representatives from all systems and stakeholders
E. Meta analysis of acquisition literature
F. Methods for managing steering based approach
G. Incentive structures
H. Rapid change management techniques (acquisition and technical points of view)
I. Tools for impact analysis
J. Fault tree/FEMA analysis to determine probability of success for complex programmes
K. Modelling and simulation for SoS programmes
L. Cost and schedule estimation tools
M. Identification and evaluation in complex SoS of best practices in this environment.

Note

[1]Committed, Reliable, Authorised, Collaborative, and Knowledgeable.