Architecting C4I Systems

ABSTRACT

DSTA has adopted the Systems Engineering methodology for developing Command, Control, Communications, Computers and Intelligence (C4I) systems. The Systems Engineering approach ensures the alignment of the systems solution to the business needs and processes of the users. It improves the processes and methodology through experimenting with and harnessing new technologies such as Business Process Management and configuration management. Systems are base-lined and fielded for operational validation and verification. Operational feedback is collected, analysed and synthesised to fill the operational gaps and meet future operational challenges. This article addresses the key issues in designing and developing C4I systems. When architecting the C4I System of Systems, the systems architect has to synthesise the C4I systems incorporating other present and future operational systems that meet the network-centric warfare concept. The article also discusses the architecture principles that ensure all C4I systems are integrated and harmonised.

Dr Yeoh Lean Weng
Ng Ming Chun

Architecting C4I Systems

INTRODUCTION

Command, Control, Communications, Computers and Intelligence (C4I) systems provide battlefield information for commanders to make decisions and control military forces to accomplish missions. DSTA leverages state-of-the-art info-communications technology to design C4I systems that bring better situational awareness to commanders. C4I systems provide comprehensive information to the commanders in a timely fashion and enable them to disseminate orders expeditiously to the troops. This will enable the ground troops to execute their missions effectively. The earlier generation of C4I systems were largely designed in a stovepipe manner for specific missions. As military operations are increasingly conducted in a network-centric manner, the methodology for architecting and developing C4I systems has to evolve to meet the System-of-Systems (SoS) capability. To meet the complex operational requirements today, C4I systems must be able to inter-operate with other weapon systems as part of a larger complex system, or the SoS. The C4I SoS has to work coherently to deliver operational capabilities that are greater than the sum of what each component system can provide.

A Systems Architecting (SA) Process was developed to guide developers in designing robust, coherent, enduring and cost-effective C4I systems that would provide network-centric warfare capabilities. Essentially, SA enables the construction of an enterprise-level architecture for the various systems to interoperate in an integrated and coherent manner in order to achieve synergistic operational capabilities. A component system can be a C4I system, a weapon system, a logistics system or an IT system. Architecting these component systems to operate coherently and to deliver the intended SoS capabilities requires a balanced application of science and art.

Rechtin suggested that SA is both a science and an art in designing and building effective and efficient SoS (Rechtin, 1997). It is a science because it uses architectural tools such as the Enterprise Architecture Framework (EAF) to capture the various perspectives of the interconnecting systems for the stakeholders to share a common understanding of the SoS. The tools and techniques also address the global integration, consistency and integrity of the SoS design. As an art, SA aims to balance the local needs of the stakeholders of the component systems in the interest of a global optimum.

More often than not, it is challenging to design an optimal SoS from analytical or mathematical derivations alone. Instead, the SoS solution is often derived from a combination of analytical exercises as well as intellectual discussions and engagements with the key decision makers and stakeholders. By leveraging the experiences of leading domain experts such as senior commanders and large-scale systems engineering practitioners, the systems architects are able to understand the utility and effectiveness of new military operational concepts and develop new operational ideas. The collaborative approach in developing the SoS architecture helps to garner greater buy-in from the users and stakeholders. This close coordination among the stakeholders ensures systems interoperability and encourages sharing of critical resources.

SYSTEMS ARCHITECTING PROCESS

The SA Process is a six-step process designed to guide the systems architects in developing the SoS architecture from systems conceptualisation to operationalisation.

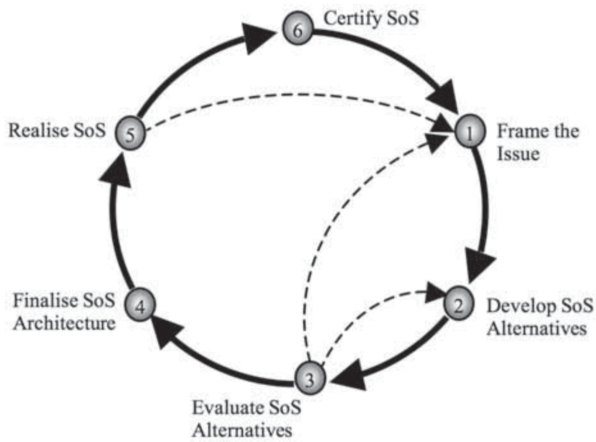

Figure 1. The SA Process

Figure 1 illustrates the SA Process. The first four steps focus on understanding the issue and developing an SoS architecture to address it. The process of framing the issue to developing the architecture typically takes about three to six months to complete. The fifth step is to implement the SoS. The time needed to design, develop, install and verify the SoS can take two to three years. The sixth step is to validate and certify that the SoS has addressed the identified issue. The SA process advocates *iterations* throughout the architecting process. This is especially important when the development of the SoS spans several years. By then, some of the operational requirements and boundaries may have changed. This may require the SoS architecture to be regularly examined for adaptation and relevance.

The SA Process exhibits a *recursive* characteristic that is able to manage the SoS complexity better. As the SoS complexity increases, SA child processes can be spun off to handle the needs at four different levels – Product, System, Capability and Enterprise.

Figure 2 shows the relationships among the parent and child SA processes. The recursive characteristic provides the flexibility for the systems architect and stakeholders to evolve system requirements, conduct experiments

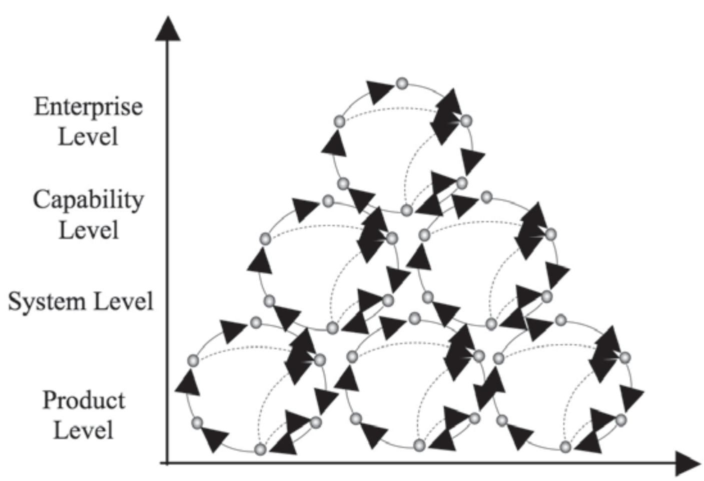

Figure 2. Multi-level application of the SA Process

with different products, and explore different approaches to the solution more effectively and efficiently. When an SA child process is completed, its capabilities are then consolidated at the parent level for eventual integration at the SoS level. For example, in an integrated air defence programme, the architecting process would be spun off into several child processes to design the command and control, weapon, communications, and surveillance systems. The resulting architecture at the system level would then be consolidated at the capability level to deliver an integrated air defence architecture.

The following paragraphs describe the details of the six-step systems architecting methodology:

Step 1: Frame the Issue. This step could influence the number of iterations of the subsequent steps before a satisfactory SoS architecture is achieved. It aims to identify and articulate the higher intent and the system needs. It will uncover the underlying assumptions, constraints and limitations to establish a comprehensive, unambiguous and accurate representation of the issue. In order to develop a deeper understanding of the issue, all the stakeholders are brought together to examine and evolve the strategic, operational and technical perspectives of the issue.

A set of techniques such as the Stakeholder Analysis, Systems Thinking, and Systems Decomposition is useful in assisting the systems architects to carry out this step. The Stakeholder Analysis is used to identify all the individuals or groups who have the potential to affect certain activities in a network-centric operation and involve them in the business analysis process. Interviews, focus group discussions and surveys as well as structured walkthroughs with the stakeholders are some useful avenues for the stakeholders to share their concerns with the systems architects. Architectural Diagrams, Use Cases and Activity Diagrams can also be used to capture the current processes and gaps in the individual component systems to help uncover constraints and limitations of the operating environment. The systems architect also applies Systems Thinking to construct an overview of the issue. This enables him to adopt a top-down approach to define the various related issues in order to enhance the understanding of the higher intent.

Capability gaps can be identified using the above techniques. These gaps will serve as input for the designers to develop the SoS requirements. A set of Measures of Effectiveness and its corresponding Measures of Performance that define how well the SoS will perform to meet the operational needs are also established at this stage for later use.

Step 2: Develop System-of-Systems Alternatives. The second step of the architecting process is to generate a broad range of possible architectural solutions to address capability gaps. The emphasis is on the exploration of the solution space and to consider solutions involving any combination of doctrine, organisation, personnel, training, system and facility. Some techniques and tools such as Modelling and Simulation (M&S) as well as Functional and Requirement Analyses are useful to the systems architects for designing different architectural solutions.

While exploring solutions, the systems architects need to understand the inherent behaviour of the SoS alternatives. This allows them to evaluate the potentiality of the SoS alternatives to address the identified capability gaps. However, it is usually very expensive to build the physical systems and to conduct field tests just for the purposes of study or evaluation, especially when weapon platforms are involved. M&S is often used to enable the systems architects to understand the architectural behaviour of the SoS at the early design stage at a relatively low cost and within a shorter period of time. A model is a simplified representation of a system in a real world. Simulation is the manipulation of the model over time and space. Using M&S allows the systems architects to tune the model's parameters easily, which represent the properties of the real systems, and re-run the simulation to learn about the behaviour of the SoS. By varying the model's parameters, the systems architects could generate different SoS alternatives to suit different degrees of operational needs such as high system responsiveness versus extreme system security.

Functional Analysis decomposes all facets of SoS capability and functionality requirements developed from Step 1 into a hierarchy of functions. At the top of the hierarchy is the enterprise level function. The enterprise level function is then successively partitioned into different sets of lower level functions or sub-functions. Further partitioning of the sub-functions may be required if its usefulness is warranted. Each set of sub-functions uses some of the inputs and produces some of the output of the parent function. Eventually, a hierarchy of operational functions of the SoS is obtained.

Requirement Analysis groups the functions or sub-functions logically into different component systems that will eventually make up the SoS. The desired performance specifications for each component system can be specified where necessary. By varying the grouping of the functions into different component systems, different SoS alternatives can be generated.

Step 3: Evaluate System-of-Systems Alternatives. The third step is to evaluate the set of SoS alternatives based on performance, robustness and cost. The aim of the evaluation is to find the most flexible and scalable architecture capable of adapting to future requirements at a reasonable cost. At this stage, software models would need to be developed to represent each SoS alternative. Some of the software models that may have been built in Step 2 can be reused here. Separately, a set of test and evaluation parameters is also established to facilitate the incorporation of test criteria into the models. Some recommended methods and tools for the evaluation of SoS alternatives include M&S, Operational Analysis (OA) and Red Teaming.

OA can be utilised to provide a quantitative assessment of the design of an SoS alternative. It uses a combination of mathematical and scientific methods such as mathematical models, probability and statistics, and algorithms to derive an optimal or near-optimal solution to complex problems. OA models such as combat models, resource/logistics models and cost models can be formulated to generate the optimal solution to address the issues identified.

Red Teaming is used for analysing systems from the perspective of a devil's advocate. The ability to analyse possible courses of action from the point of view of an adversary helps the systems architects to identify the architectural vulnerabilities and rule out the SoS alternatives. Alternatively, they may tweak or re-design the SoS alternatives to mitigate or eliminate the vulnerabilities.

During the evaluation process, new insights from the analysis may emerge, and these result in the need to re-define the issue or to refine the SoS design. It may be necessary to go through several iterations before a satisfactory architectural solution is derived.

Step 4: Finalise System-of-Systems Architecture. The output of SA is an endorsed SoS architecture. This SoS architecture is described in terms of architectural views in accordance with the EAF and governance guidelines. The documentation will facilitate the promulgation, communication, masterplanning and realisation of the SoS architecture.

The finalised SoS architecture will facilitate the formulation of the various masterplans that will chart the milestones for capability build-up, resource planning and competency development.

Step 5: Realise System of Systems. The SoS is then built based on the endorsed architecture. Given the size and complexity of a typical SoS, its implementation is usually broken down into several component systems for better management. There will be different programme teams responsible for the acquisition and development of various component systems. Where necessary, a Technical Working Group or Programme Steering Committee may be formed to provide management guidance to the programme teams. The need for comprehensive planning and close collaboration among the various programme teams should be emphasised. While each of the programme teams is likely to have its implementation schedule and priorities, there will be a master schedule to keep tabs on the progress of each component system. This will enable component systems integration and capability demonstrations to be carried out at scheduled milestones, and also serve to track the progress of SoS integration.

During the SoS realisation stage, any deviation of the SoS architecture will need to be raised at appropriate governance forums for endorsement. Since the realisation of the SoS may take several years, it is possible that the changing external environment may invalidate the assumptions made during the architecting process. This may result in the need to re-examine the SoS architecture's relevance to the new environment.

Step 6: Certify System of Systems. Verification, validation and certification of the SoS are essential activities during this process. The SoS will be evaluated and validated for its capability and performance with respect to the masterplan. Verification is a quality management process to ensure that a system complies with specifications and should be conducted throughout the systems development phase. Validation is used to establish a level of confidence that a system accomplishes its intended mission capabilities and addresses user needs. When the SoS is successfully verified and validated with respect to the masterplan, the systems architects can proceed to certify the SoS with the customers and stakeholders.

The different capability build-up plans for various component systems will lead to several SoS capability milestones. Since the SoS consists of multiple component systems, end-to-end testing is generally costly and time-consuming, especially when weapons platforms are involved. Therefore, M&S developed in an SoS Integration Lab would serve as valuable surrogates to emulate end-to-end SoS validation. With the aid of M&S, key test points can be selected to validate the SoS performance. The process of verification and validation may lead to new insights or the discovery of unexpected emergent behaviour. It is therefore important to perform regular monitoring of SoS operations to look out for emergent behaviour that may warrant adaptation to the architecture. If changes in the external environment result in deficiencies in the SoS architecture, it may be necessary to re-examine the relevance of the SoS architecture and re-initiate the whole architecting process.

ENABLING THE ARCHITECTING OF C4I SYSTEMS

As armed forces adopt a network-centric approach to conduct military operations, C4I systems have to be designed based on the SoS concept where the C4I systems of the different stakeholders are integrated horizontally so that all the command posts are interoperable by design. The network-centric warfare concept marks a departure from the traditional command-centric approach in which the stakeholders operate more independently in a mission. Thus, when the network-centric warfare concept was introduced, our stakeholders had to align quickly with this transformation. They were faced with a critical need to articulate the transformed operational requirements to support this new concept of operations.

To be clearer about their needs, stakeholders and technologists were brought together through meetings, forums, working groups and focus groups to share and discuss their suggestions and proposals, voice their concerns, and brainstorm new ideas. These discussions have helped them to achieve clarity on their needs.

The key stakeholders were identified and grouped according to their seniority and professional domains for better management. The senior commanders were first consulted to understand the strategy of conducting future military operations with a system of fully networked fighting forces. This vision helped to provide guidance for discussions at the lower levels. Some stakeholders were also grouped into different domain groups such as airbase operations, logistics or human resource to address specific operational needs that would be required to support network-centric operations. The EAF was used during this 'frame-the-issue' stage to translate the operational vision of the senior commander into specific systems requirements. Based on the needs gathered from different levels and domains, the technologists were able to quickly produce an initial hierarchical structure of system components and functions by using the System Decomposition technique. The architectural views of the EAF were used to capture the operational, system and technical perspectives of the desired SoS. This is for the purpose of promulgation and communication in order to achieve synchronous understanding among the stakeholders and technologists, as well as guide the SoS development at the later stage.

Architectural Views in the EAF

There are five types of architectural views in the EAF designed to describe architecture from different viewpoints:

- The **Strategic View** describes the current capabilities, gaps, dependencies, and options available to close the gaps. It also describes the desired end state, the systems capabilities required and the strategy or approach to acquire the capabilities and achieve the end state.

- The **Service View** defines the business, technical and non-functional services required to provide a capability.

- The **Operational View** describes the operational processes and activities, the information exchanged among operational units and the outcomes of the missions.

- The **Technical View** describes the standards and the products used for the systems implementation.

- The **System View** describes the interfaces among systems, physical deployment of systems, system functions, the information exchanged by these functions, the mapping of the functions over the operational nodes, and the data model at the system level. The System View helps to identify the new system functions to be built and those to be enhanced or reused.

Indeed, when acquiring or developing C4I capabilities, the EAF advocates the reuse of software components that were previously built and tested for operational deployment and trial wherever possible. If reuse is not possible, the project team will develop new components. The project team would then contribute the components to the common repository for further reuse by other project teams. The objective is to reduce the time needed for systems development so that new capabilities can be fielded rapidly. The common repository and Service-Oriented Architecture (SOA) were put in place to allow us to build upon the intellectual capital to develop and field our C4I systems more rapidly. The Business Process Management System (BPMS) is adopted to model the users' workflow in order to further reduce the time-to-fielding and to provide operational agility by complementing SOA.

Common Repository

The common repository keeps the system business applications and technical component services that developers can draw upon to rapidly assemble and deploy C4I systems. As the repository applications and services are thoroughly tested for operational deployment, the assembled C4I systems can achieve a high degree of assured quality for operational trial and deployment. The common repository is an enterprise asset that must be properly maintained, continually expanded in the number of reuseable components and evolved through a rigorous quality management process. If new applications and services need to be developed to meet new operational requirements, they will be developed in addition to the C4I baseline systems. The C4I systems are then fielded for experimentation and exercises where the operational users will verify the functionality and validate the systems. Once validated, the new applications and services will be enhanced and tested in terms of robustness and reliability. Upon acceptance by the technical architecture working group, the new applications and services are formally incorporated into the next baseline C4I systems and added to the common repository for reuse and future development.

Service-Oriented Architecture

SOA shortens the time needed for C4I systems development by reusing existing services. It allows business logic to be exposed as services to an external application or service. It serves as a development and integration platform for the rapid development of new business applications or services by assembling the necessary services. Specifically in the area of enterprise systems integration, SOA helps to shorten the time required for C4I systems development by reusing existing services, much like reusing software components from the common repository. SOA is often implemented with the use of Web Services. It utilises ubiquitous standards or protocols for data exchange and communication such as XML and Internet Protocol. The use of common standards and protocols eases the task of integrating heterogeneous systems in a network-centric environment. Figure 3 shows a multi-tier C4I systems architecture on which our C4I systems are built. The business logic layer shows a simplified view of the SOA and the range of services that was incrementally built up over the years.

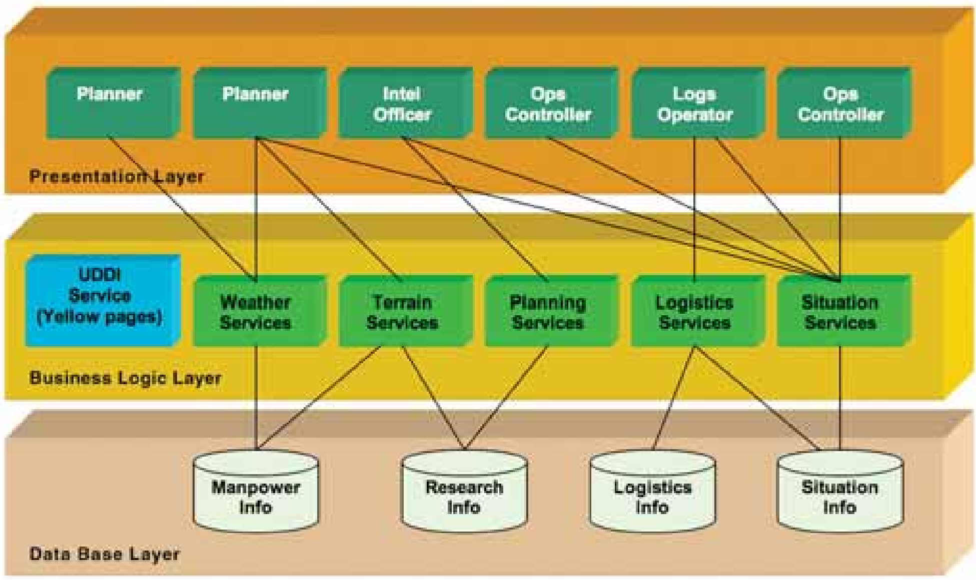

Figure 3. Multi-tier C4I systems architecture

Business Process Management System

In SOA, the business services support the business processes and these are manifested in the business logic layer of the multi-tier architecture. Changes in operational requirements could be addressed through modification in the business logic. In a fast-changing environment, business processes will need to continually evolve to meet new challenges. However, managing business processes is a big challenge. BPMS is the IT enabler for managing processes from process definition to retirement. Most commercial BPMS solutions support rapid graphical modelling, development, testing, simulation, optimisation, deployment and monitoring of the processes. BPMS enables new operational capabilities that involve the flow of tasks among different parties to be developed and deployed faster as compared to traditional development through coding. BPMS, used in combination with SOA, will further shorten the development time of a new capability. Some process-oriented capabilities can be developed by systems configuration or by assembling existing software components and services in the SOA. As BPMS externalises the business logic from the software code, the changes to the processes can also be implemented easily and deployed on the fly. This gives the much-needed operational agility to the stakeholders as they operate in a very dynamic environment. BPMS is an emergent and highly promising technology that is currently being explored for C4I systems development.

While the EAF is used to guide the design of C4I systems and various technologies like SOA are used in the rapid development of C4I capabilities, it is essential that proper governance and competency development frameworks are put in place to ensure that systems are built according to established standards and are interoperable by design.

IT Governance

In DSTA, a technical working group was set up to ensure that the architecture and technical solutions of a system comply with the reference architecture or the approved SoS architecture. As part of DSTA's Quality Management System, every C4I system project is required to seek the technical working group's endorsement for its solutions prior to the start of its development. The project manager is required to submit the solution in the form of system and technical views. The working group also scrutinises the operational views to ensure the alignment of the system solution to the business needs and processes of the users.

Competency Development

As technology advances by leaps and bounds, it is important for an organisation to invest continually in their human resource by upgrading the technologists' technical skills and developing new competencies. The effort includes identifying emerging competencies needed for future challenges and committing resources, time and effort to develop the technologists. The Directorate of Organisation Capability Development works with each technologist to come up with the annual learning plan that specifies his training needs, and the schedule and resources required to build up his skills and competency. A systems architect would need to be trained in diverse domains of technical and business competencies. He should be competent in complex systems design, systems thinking and business analysis. The 'C4I Systems Architecting Guidebook' and the 'C4I Development Guidebook and Competency Portal' in DSTA's intranet are good sources of information to begin with. The guidebooks contain a wealth of architecting knowledge that was accumulated and compiled by a group of experienced systems architects who have been through many rounds of C4I systems development. The portal serves as an online avenue for the C4I community to exchange views and share their experiences in C4I systems development.

CONCLUSION

The network-centric warfare concept elevates the need for systems interoperability to an unprecedented level. In the modern world where technology advances at blazing speed, new threats emerge faster than ever before. This drives the demand for new counteracting capabilities to be deployed at an almost equal rate. Methodologies for architecting C4I systems need to be robust enough to field systems in the shortest time possible. The search for alternatives to harness state-of-the-art technologies for the rapid development and fielding of new operational capabilities is a perennial task for C4I systems architects.

REFERENCES

Cebrowski, A., and J. Garstka. 1998. Network-centric Warfare: Its Origin and Future. Proceedings of the Naval Institute.

Jamshidi, Mo. 2009. Systems of Systems Engineering – Principles and Applications. CRC Press.

Rechtin, E. 1997. Systems Architecting: Creating and Building Complex Systems. Prentice Hall.

Rechtin, E., and M. Maier. 2000. The Art of Systems Architecting. CRC Press.

Yeoh, L.W. and L.P. Chew. 2004. System Engineering Approach to Development of C2I Systems in the Knowledge-Centric Age. Paper presented at the Massachusetts Institute of Technology Engineering Systems Symposium, 29–31 March, in Cambridge, USA.

Yeoh, L.W., H.B. Syn, and C.V. Lam. 2007. An Enterprise Framework for Developing Command and Control Systems. Paper presented at the 17th Annual International Symposium of the International Council on Systems Engineering, 24–28 June, in San Diego, USA.

Yeoh, L.W., T.L. Teo, and H.L. Lim. 2008. Continuous Systems Development for Command, Control and Intelligence Systems. Paper presented at the 18th Annual International Symposium of the International Council on Systems Engineering, 15–19 June, in Utrecht, the Netherlands.

This article was first presented at the 2nd International Symposium on Engineering Systems held from 15–17 June 2009 at the Massachusetts Institute of Technology in Cambridge, USA and has been adapted for publication in DSTA Horizons.

BIOGRAPHY

Dr Yeoh Lean Weng is Director (DSTA Masterplanning and Systems Architecting). He is concurrently the Deputy Director of the Temasek Defence Systems Institute and an Adjunct Professor at the National University of Singapore (NUS). Lean Weng has extensive experience working on large-scale defence engineering systems. As a systems architect, he played a key role in developing the Enterprise System of Systems Architecture for defence applications. Lean Weng also developed the systems architecting methodology for masterplanning and defence transformation. He is the President of INCOSE Singapore Chapter, the INCOSE Region VI Representative to the Member Board and an INCOSE Fellow. He is the Vice President and Chairman of the Industrial Group at the Institution of Engineers, Singapore (IES). He is a founding member of the IES Certified Systems Engineering Professional and Certified Project Managers Registry. Lean Weng received his Bachelor and Master of Science degrees from NUS in 1983 and 1987 respectively. He further obtained two Master degrees in 1990 and a PhD in Electrical Engineering from the Naval Postgraduate School, USA in 1997.

Ng Ming Chun is a Senior Engineer (C4I Development). He is presently leading a Command, Control, Communications, Computers and Intelligence development team, as well as pursuing his interest in Systems Architecting and Business Process Management System. Ming Chun researched on distributed systems and published a paper on 'Adaptive Home-based Software Distributed Shared Memory System' in the International Conference on Parallel and Distributed Systems 2000. He obtained his Bachelor (Honours) and Master of Science degrees in Computer Science and Information Systems from NUS in 1997 and 1999 respectively.