

# A Program of Research into Systems Engineering

Joseph E. Kasser DSc CEng CM MIEE, Stephen C. Cook PhD FIEE FIEAust  
Systems Engineering and Evaluation Centre  
University of South Australia  
School of Electrical and Information Engineering  
Mawson Lakes Campus, South Australia 5095  
Telephone: +61 (08) 8302 3941, Fax: +61 (08) 8302 4723  
Emails: [Joseph.Kasser@unisa.edu.au](mailto:Joseph.Kasser@unisa.edu.au), [Stephen.Cook@unisa.edu.au](mailto:Stephen.Cook@unisa.edu.au)

**Abstract.** This paper provides an overview of a number of research areas that include investigating the nature of systems engineering and its underlying concepts, defining the properties of object-oriented requirements, producing prototype object-oriented tools for systems engineering, and applying of systems engineering to various domains.

## Introduction

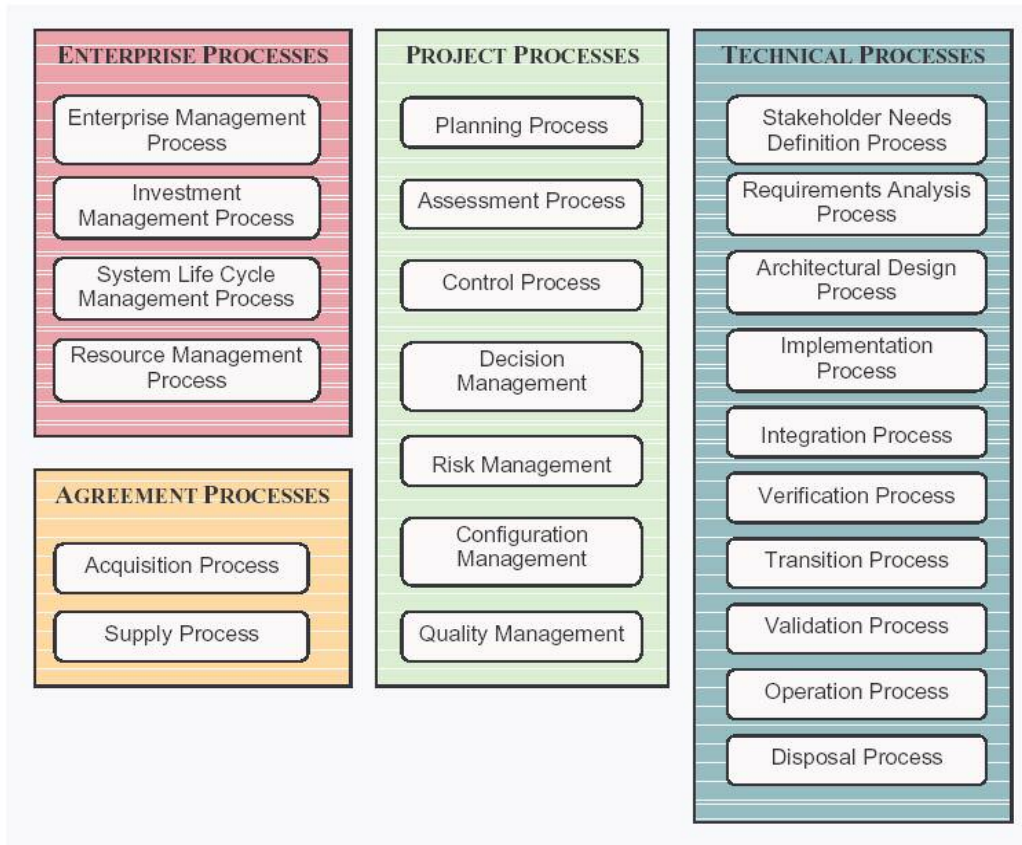
This paper provides an overview of the research performed into and about systems engineering at the Systems Engineering and Evaluation Centre (SEEC) in the University of South Australia (UniSA). The specific areas of research summarised in this paper cover:

- The elements of a framework for the engineering of complex systems;
- Object-oriented systems engineering;
- Object-oriented requirements;
- Research into prototype object-oriented tools for systems engineering;
- The application of systems engineering to various domains.

## ***Elements of a Framework for the Engineering of Complex Systems***

This area of research attempts to identify the principal elements of a framework for the engineering of complex systems. Systems engineering, the creation of large complex, technical systems, has been a recognised activity for over fifty years. Over most of this time, systems engineering has been considered as a practice-based activity rather than a discipline in its own right. This perception has been changing over the last 15 years since the genesis of professional societies such as the International Council on Systems Engineering (INCOSE). There are now over 100 postgraduate programs in the field, catering to an ever-growing demand (Fabryky, 2003). The Education and Research Technical Committee of INCOSE has been establishing a body of knowledge for systems engineering (Leibrandt, 2001) that can be used to inform teaching at universities and training needs within a workplace-based employee competency framework. The authors have contributed to this work (Kasser and Massie, 2001) and have accepted the challenge to work as part of an international working group to establish a framework for research into the discipline of systems engineering.

The recently released ISO/IEC systems engineering standard (ISO/IEC 15288:2002) provides a framework for a programme of research into systems engineering. The standard is the newest and highest level systems engineering standard to be published. ISO/IEC 15288 covers a broad area of processes mapped into four groupings as shown in Figure 1. The importance of ISO/IEC 15288:2002 is that it encompasses all the activities in the earlier standards (such as ANSI/EIA-



**Figure 1. The system life cycle processes (ISO/IEC 15288: 2002; p61) provides the framework for the research program.**

632, IEEE 1220:1998, MIL-STD-499B) and explicitly adds the enterprise processes that (ISO/IEC 15288:2002: p8-9):

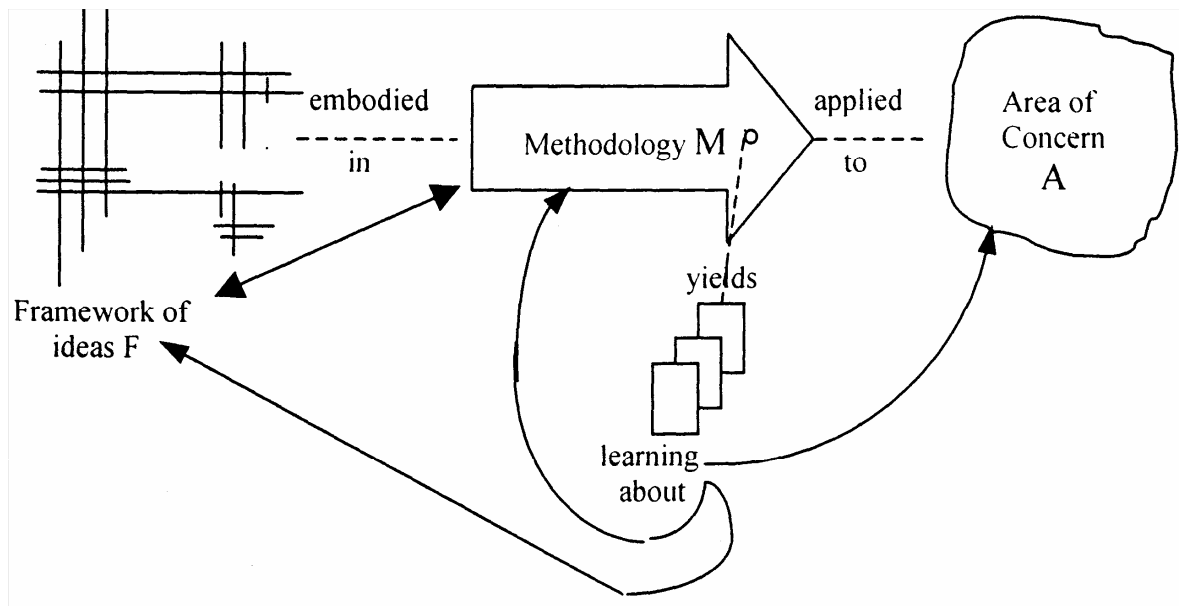
“manage the organization’s capability to acquire and supply products or services through the initiation, support and control of projects. They provide resources and infrastructure necessary to support projects and ensure the satisfaction of organizational objectives and established agreements.”

Checkland and Holwell (1998), state that there are three elements necessary to describe any piece of research. These are:

- **The Area of Concern (A)**, which might be a particular problem in a discipline (area of study), a real-world problem situation, or a system of interest.
- **A particular linked Framework of Ideas (F)** in which the knowledge about the area of concern is expressed. It includes current theories, bodies of knowledge, heuristics, etc as documented in the literature as well as tacit knowledge.
- **The Methodology (M)** in which the framework is embodied that incorporates methods, tools, and techniques in a manner appropriate to the discipline that uses them to investigate the area of concern.

Figure 2 extracted from Checkland and Holwell (1998), illustrates the relationship between these three elements and how undertaking the methodology creates new knowledge about all three elements. These same three elements can also be used to characterise a discipline because they encompass the key aspects of a discipline: a specific area of study (A), a literature (F), an agreed methodology (M), given that there is a working community of paid scholars and/or practitioners, (Kline, 1995, p3).

The research approach is to identify appropriate areas of concern, methodologies, and frameworks of ideas appropriate to systems engineering. The first avenue being explored on this journey has shown that systems engineering can be thought of as a multi-methodology that applies to the following two areas of concern (Cook et al., 2003; Ferris et al., 2003):

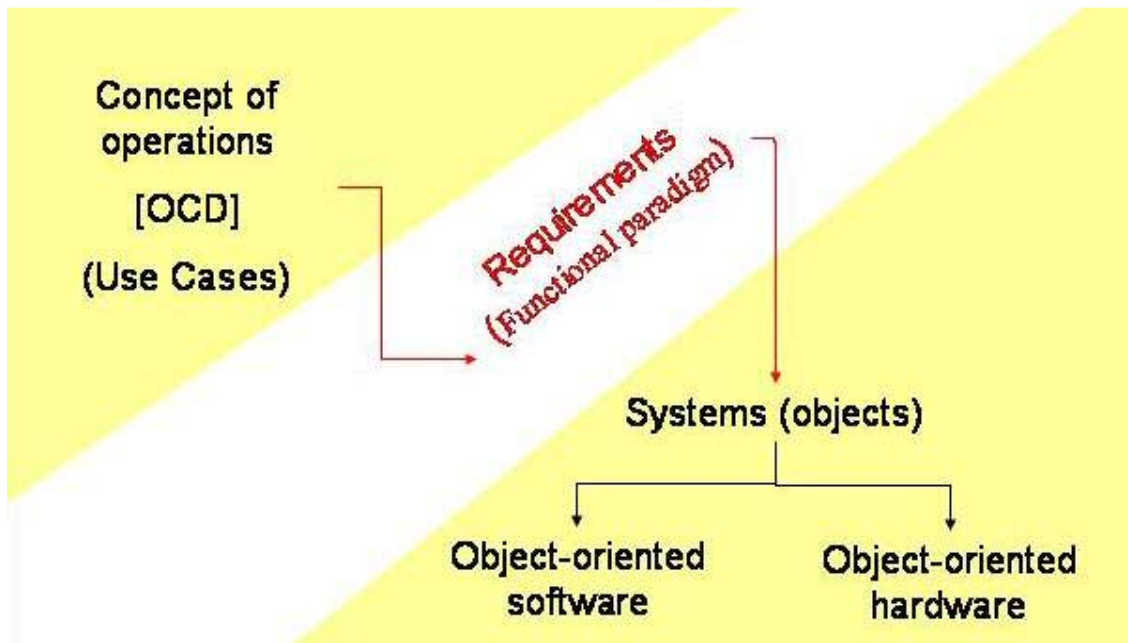


**Figure 2. Elements relevant to any piece of research (Checkland and Holwell, 1998: p 13).**

- A designed physical system (that also contains human components).
- A human activity system (engineering organisation).

The two very broad areas of concern identified span many disciplines and require pluralistic approaches that not only invoke multiple methodologies but ones that rest on quite distinctly different frameworks of ideas. This observation indicates that the two areas of concern are of fundamentally different types and hence will probably have to be approached with different methodologies. As such we now understand why:

- Many systems engineers cannot clearly articulate the functions and benefits of systems engineering (Kasser and Shoshany 2001).
- It has been extremely difficult to establish a systems engineering body of knowledge (SEBOK) for the diverse activities that are known by the term “systems engineering.” (Kasser and Massie, 2001).



**Figure 3. The gap in Object-oriented Systems Engineering.**

### ***Object-oriented systems engineering***

In the document-driven paradigm of the 20<sup>th</sup> century, the process-products of systems engineering were paper based. Each process-product was produced by means of a sequential process. The 21<sup>st</sup> century paradigm is based on electronic databases. Early renditions of these electronic databases have inherited the document approach of separate databases, resulting in separate and distinct databases for operations concepts, requirements, and project management. The research program into object-oriented systems engineering considers the systems life cycle from an information management perspective and integrates the necessary information about the product and process into a single conceptual database. In this approach, the various process-products are abstracted views of the single conceptual database. Thus an Operations Concept Document (OCD) becomes one type of abstracted view, a requirements document a second, and a test plan a third, etc. This concept of a single virtual database for project information forms the underpinning thread of SEEC's research program into object-oriented systems engineering.

Several approaches to object-oriented systems engineering have been proposed (e.g. Hopkins 1998; Meilich 1999; Lykins 2000) and there is an INCOSE working group working on upgrading the UML to add "systems" aspects. However, Kasser (2003) identified a disconnection in the object-oriented systems engineering process as depicted in Figure 3. Concepts of operations are stated in the form of Use Cases involving the interaction of objects, and the system (the object of the production process) is developed to perform the functionality described in the Use Cases. Requirements, however, remain firmly in the functional paradigm as these object-oriented efforts, in general, have not applied the object-oriented concept to requirements. The general approach seems to be to either:

- Treat requirements as something that has to be produced in the early stages of the System

Life Cycle, and sometimes, add properties of traceability and priority. Schach (2002) for example, still partitions requirements into functional and non-functional, but does add the properties of traceability and priority.

- Ignore requirements other than those that can be expressed in Use Cases. This approach in general just seems to articulate the user's needs in two redundant ways (Use Cases and requirements).

So from a process perspective the front end of the process (concept of operations stating the customer's need) is object-oriented, the back end of the process (the implementation of the system to meet the customer's needs) is also object-oriented but the requirements remain in the functional paradigm. Kasser and Schermerhorn (1994) wrote that systems engineers needed to use a methodology that seamlessly interfaced to the software development methodology to avoid delays and errors in translation from the system requirements to the design phase of the SLC. Consequently, there is a need for object-oriented requirements in an object-oriented implementation paradigm.

### **Object-oriented requirements**

According to Kuhn (1970), a change in paradigm provides a marked improvement over the earlier paradigm and tends to resolve paradoxes in that paradigm. The systems and software development industry is characterized by a paradigm of project failure (Standish 1995). The situation has been described by Cobb's Paradox (Voyages 1996), which stated "*We know why projects fail, we know how to prevent their failure --so why do they still fail?*" One of the known contributing causes of these project failures is poor requirements engineering and management, which has been repeatedly and widely discussed and documented for at least 10 years (Hooks 1993; Kasser and Schermerhorn 1994; Jacobs 1999; Carson 2001; etc.). However, this continual documentation and discussion of the problem of poor requirements engineering and management has not resulted in a practical solution to the problem. This research is to develop an object-oriented approach to requirements engineering that might help to reduce the contribution of poor requirements engineering and management to project failures, namely resolve part of Cobb's Paradox.

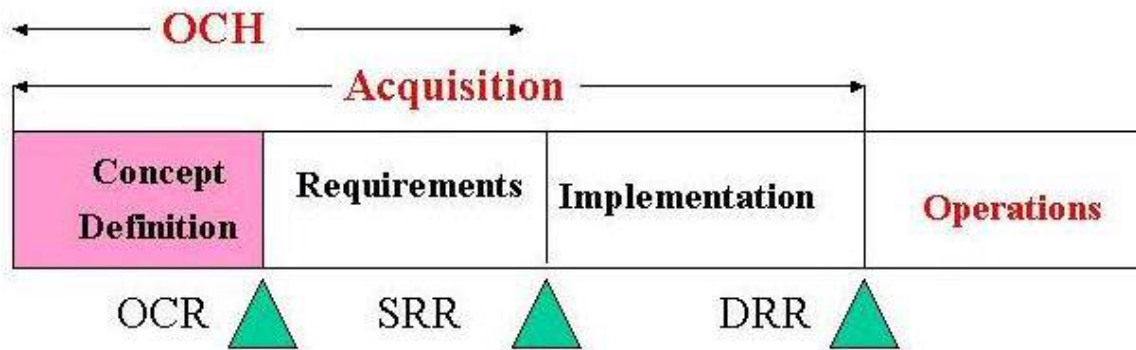
Systems engineering is focused on dealing with well-structured problems. However, the problem of poor requirements is a complex and ill-structured problem, and hence not solvable by the traditional systems engineering process. One approach to dealing with complex and ill-structured problems, such as the problem of poor requirements, is through the realisation that an accumulation of knowledge about the system can often clarify the problem (Nii 1986). The major research questions in this area are:

- Can the object-oriented approach eliminate the need for text-based requirements?
- What are the properties of an object-oriented requirement?

The answers are not simple. Consider the general approach, and then each question. This research has taken the form of an evolutionary process for the continued evolution of knowledge management capabilities using an approach named "Rapid Incremental Solution Construction" (RISC)<sup>1</sup> (Kasser and Cook, 2003). It is

---

<sup>1</sup> The authors are very conscious of the risks involved in implementing partial solutions to a problem; hence the choice of acronym.



Milestone reviews

**Figure 4. The Place of the OCH in the System Life Cycle.**

- Based on the Blackboard methodology pioneered by Nii (1986);
- Agent based, by virtue of using a suite of software products for rapid software evolution as a way to develop software that is quicker and less expensive to maintain (Glover and Bennett 1996);
- An incremental multi-disciplinary approach based on the Cataract Methodology (Kasser 2002);
- Reductionist, namely to first provide a solution for a part of the problem, and then provide a solution for another part, and so on, until the entire problem (or at least a major part of it) is eventually solved;
- An approach to solving the problem of poor requirements by developing a collaborative system that includes encapsulated tools, newly designed custom software, and human collaborators (Lander 1997);
- Capable of providing concept demonstrators of the functionality needed in the form of a suite of simple prototype next generation Computer Enhanced Systems Engineering (CESE) tools or software agents having similar user interfaces and each performing a small range of functions.

We now consider each of the research questions in turn.

### ***Can the object-oriented approach eliminate the need for text-based requirements?***

This research program is based on the recognition that **requirements are a means, not an end**. There is nothing divine about requirements; they are just a convenient poorly-used tool for translating user needs for the boundary of the design space of the system that should be built. Gabb et al. (2001) define a requirement as *“an expression of a perceived need that something be accomplished or realized.* Van Gaasbeek and Martin (2001) quote Dahlberg as stating that **“we don't perform systems engineering to get requirements and add “we perform systems engineering to get systems that meet specific needs and expectations.”** The focus is on user needs, not requirements. Requirements are developed as an intermediate work product in the system development process, and are developed to provide formal communication between the

stakeholders. Writing text-based, unambiguous requirements for combinatorial and sequential scenarios in the form of imperative construct statements is difficult. Timing and state diagrams are often used within the context of the Requirements Document to provide the necessary information. Thus the concept of stating user needs (under certain circumstances) via diagrams is already in use in systems engineering. Sutcliffe et al. (1999) proposed reducing human error in generating requirements by analysing requirements using an approach of creating scenarios as threads of behaviour through a *use case*, and adopting an object-oriented approach.

Our research program examined system engineering and object-oriented methodologies and determined that both systems engineering is inherently object-oriented and that object-oriented languages such as the Unified Modelling Language (UML) may be used to document the user's needs in a manner that can be used by developers. This thread of research led to the development of a prototype new generation Computer Enhanced Systems Engineering (CESE) tool known as an Operations Concept Harbinger (OCH) (Kasser, Cook, et al., 2002) that could be used to hold both user and developer representation of the user's needs as an alternative to, and an improvement on, text mode "requirements", hence increasing the reliability of the shared meaning of the user's needs amongst all stakeholders (Kasser 2002a).

The OCH bridges the gap between the soft systems methodologies used in the early phases of the system development life cycle to elicit and elucidate requirements and the hard systems methodologies used in the construction of a system. The OCH may be thought of as a multimedia Operations Concept Document that also contains measures of effectiveness for each operational scenario. The OCH has the potential to:

- Improve the requirements elicitation process, by providing an interactive facility for exploring the capabilities of the customer's needs;
- Minimize the effect of poorly written requirements by using an object-oriented approach to tag *use case* scenarios in the concept of operations with measures of effectiveness and acceptance criteria (in response to the question "how will you know when the system of interest meets your needs?");
- Improve the acquisition and life cycle management of both single and multiple systems. While suitable for representing single systems, the OCH will be a better tool for representing Systems of Systems. OCDs and SRDs can be bulky for single systems, while the combined OCD and SRDs for multiple systems pose a daunting amount of reading that can only be managed effectively using some kind of CESE tool.

The OCH is a logical extension of the tools that are readily available for building and storing UML descriptions of systems. Consider the situation in which UML would be used for the *use cases* and tags would be used to contain Measures of Effectiveness (MOE) for the *use cases*. Thus not only would the tool contain the user needs in the form of *use cases*, but the conditions for showing how the user needs would be deemed to have been met in every *use case* would also be built into the tool. The designers would then have the need or "requirement" to design to, and the Test and Evaluation (T&E) staff would have the necessary information to define a Test and Evaluation Master Plan. This approach is the functional equivalent of combining the OCD and System Requirements Document (SRD) in a tool database (together with other data). It would eliminate the need to produce requirements documents and thus bypass the generation and effect of vague and unverifiable requirements in the current paradigm.

Figure 4 shows that the use of the OCH occurs during the needs elicitation process and the requirements documentation sections of the system life cycle. The OCH provides capability that

is currently provided by text mode OCDs, SRDs, PERT and GANTT charts, some simulations and some prototypes. Since the OCH may contain animated graphics and various work flow charts for describing processes, the OCH will be cheaper to implement than simulations for multiple systems scenarios.

When the contents of the OCH are displayed, an interactive dialogue takes place between designer and customer that clarifies the scenarios and can add the MOE. This feature helps bridge the gap between Soft Systems methodologies (Checkland 1993) used to elicit user needs for the system of interest and the hard systems methodologies used to construct the system. Any appropriate workflow model or analysis methodology can form the basis for the analysis and exploration of the user needs and the transformation into the *use cases*.

The OCH has the potential to improve the way Systems of Systems are managed. The prototype has shown that in the concept definition stage, the requirements elicitation and identification process is enhanced by the multimedia technology embodied in the OCH (Kasser, Cook, et al, 2002).

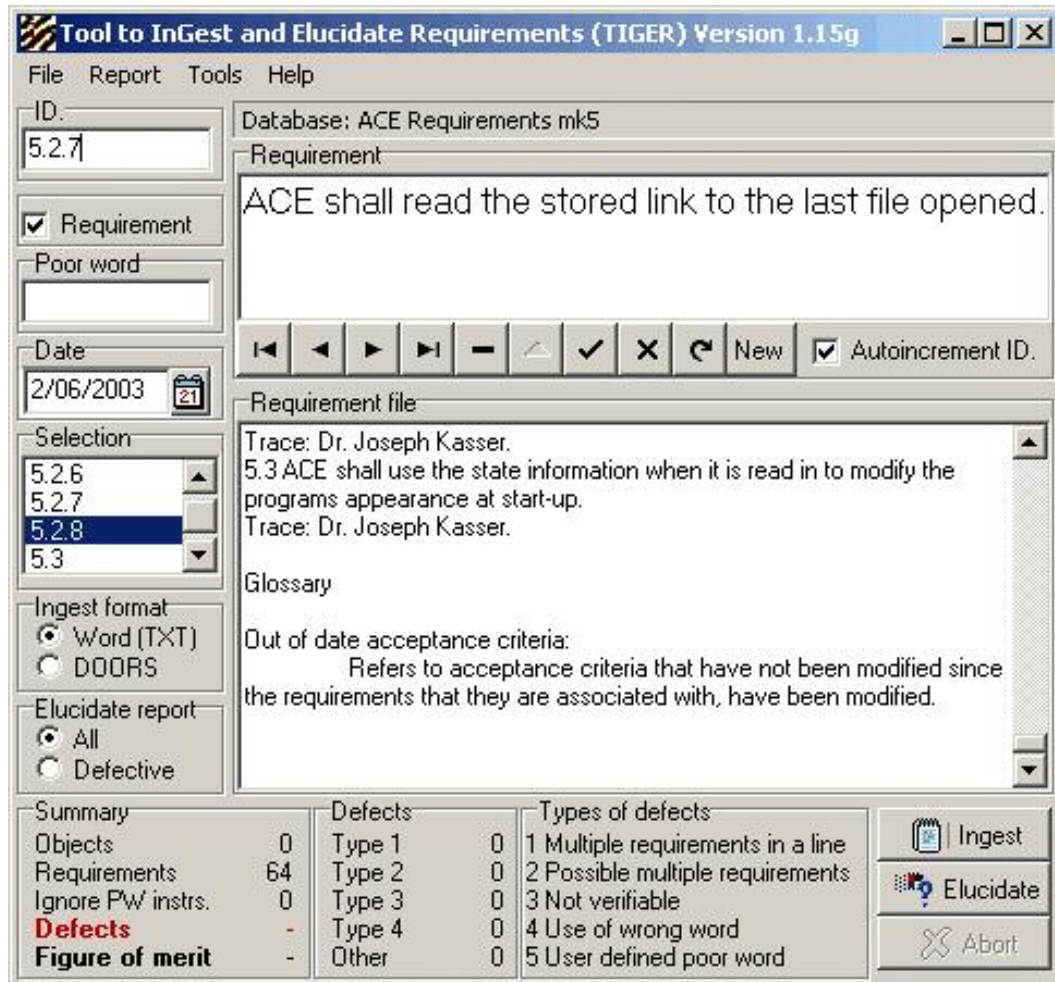
### ***What are the properties of object-oriented requirements?***

While the OCH does not use “requirements” per se, it is an object-oriented tool and contains information about the target system and production process. This information could be considered as taking the form of a subset of the properties of object-oriented requirements. Before attempting to identify the properties of object-oriented requirements, a set of rules were to be established based on the maxim that a good requirement has the following three characteristics:

1. **It describes something about the physical system that will meet the needs of the customer.** This is the traditional text based sentence that covers the functional and non-functional aspects of the system being produced.
2. **It facilitates (or rather not does not impede) the production process.** This characteristic is derived from Total Quality Management (defined by NASA (1992) as the application of systems engineering to the work environment) and is concerned with the effectiveness of the production process. While requirements define a need, they can also be viewed from the contractual perspective. The cost of realising a system is based on transforming requirements into systems. From the contractor or supplier’s perspective, requirements translate as profit on the contract. The contractor’s goal is to maximise profit, while the customer’s perspective, is aimed at purchasing at the lowest whole life cycle cost to build, without compromising conformance to specifications (Crosby 1979). Defects are not free, someone makes them, and gets paid for making them (Deming 1986, p11) and then someone pays for them to be repaired. Properties based on this characteristic include vagueness and ambiguity, namely properties that lead to cost escalations, schedule delays, or the provision of undesired functionality.
3. **It is something the customer really wants.** This is the most difficult characteristic since customers do not always to state the requirement.

The first avenue being explored on this journey to identify the properties of an object-oriented requirement is the usage of requirements in the System Life Cycle. Early research based on the object-oriented concept, Total Quality Management, and a review of the requirements engineering and systems engineering literature, has identified:

- a set of Quality System Elements that form candidates for properties of an object-oriented requirement (Kasser 2003);
- some functionality that can be incorporated into the requirements object (Kasser 2003); and
- a set of requirements for writing the wording of requirements (Kasser 1995).



**Figure 5. Main Screen display from the Tool to Ingest and Elucidate Requirements (TIGER).**

In order to investigate and determine the usefulness of these candidate properties and functionality, a set of prototype educational tools for teaching systems and software engineering (PETS) have been developed and used in postgraduate classrooms. These tools act as agents performing some functionality on a prototype object-oriented requirements database based on the candidate properties. Kasser (2002b) described the First Requirements Elucidation Demonstrator (FRED), a prototype software tool that scanned the wording of requirements and flagged any requirements that were poorly written or did not comply with the requirements for writing requirements (Kasser 1995). An updated tool to ingest and elucidate requirements (TIGER), shown in Figure 5, has been used in a class lecture on requirements engineering in three postgraduate courses. Before TIGER was introduced, the discussions in the tutorials focussed on

the structure and format of requirements. After TIGER was introduced and used to elucidate sample requirements, the focus of the in-class discussions changed to cover the difficulties of writing good requirements. This is a significant shift in perspective (Kasser et al., 2003).

Another area of research extends the use of requirements engineering tools to assist the writers of requirements to only accept requirements that are feasible using the Frame-Based Requirements Engineering Tool (FBRET) approach (Cook et al, 2001). The approach is illustrated by the construction of the Communications Requirements Evaluation & Assessment Prototype (CREAP). The CREAP is a prototype software package that is able to integrate military communications domain expertise and requirements engineering practice in order to ensure the feasibility of equipment selected to meet the requirements imposed on communications systems for a rapid force deployment in a design to inventory scenario (Kasser and Cook, 2002). Unlike a traditional development where equipment is designed to fulfil a need, the task force information system problem can be characterized as an unprecedented, real-time component integration problem. (The term unprecedented refers to the fact that many of the parts will never have been integrated before.) The unprecedented nature of the problem means that there would be a dearth of experience from which to draw to avoid integration problems.

Like any systems design problem, task force information system design starts with a user needs statement. The needs of the user can be obtained from planning documents but many will be mission-specific for missions that may or may not resemble the proposed rapid deployment about to take place. The role of CREAP-like tools is to assist a knowledgeable user to translate these high-level user needs into a system design that can be implemented from available components (inventory). To do this task, the tool needs to be aware of the contents of the inventory and the interfacing potential of each piece of equipment and be able to make decisions as to the suitability of the combination of components for the desired mission. The CREAP is an instance of such a tool for the purpose of formulating the communications for a rapidly deployed task force using communications components in inventory. It also needs to reason using a communications (and information systems) architecture model that can estimate the performance of the proposed information system. The FBRET implementation of the CREAP is an extension of expert system programming techniques based on combining work by (Cook 1990, 1991, 1993), and (Kasser 1992).

The CREAP implementation of a FBRET using interpreted frame based knowledge and inference provides a very flexible and readily testable tool that provides a capability that is new. This research has shown that requirements engineering tools can be extended by the FBRET approach to provide useful functionality.

### ***The application of systems engineering into various domains***

The following are examples of postgraduate research performed at SEEC.

**An Acquisition Methodology for the Procurement and Integration of a Network Enabled Warfare System of Systems.** As Australia develops a Network Enabled Warfare System of Systems it is necessary to ensure that acquisition methodologies are suitably modified to allow rapid procurement of new systems and integration with legacy systems. The research investigates a novel acquisition methodology to speed procurement and integration for the Network Enabled Warfare System of Systems; achieving this through rapid prototyping, experimentation and interface engineering for legacy systems.

**A Systems Approach to the Evaluation of Complex, Real-Time Information Systems: Determining the Effectiveness of Command and Control Organisations.** This research examines the reasons for the failure of conventional T&E to validate complex, near real-time information systems (ISs) effectively and investigate the utility of other methodologies for determining the effectiveness of these ISs. The research topic considers the specific case of complex, near real-time ISs used at the military operational level of command, and thus will require a good understanding of the nature of military command and control (C2), and command support systems to be developed. The aim of this research is to propose a framework for evaluating C2 organisations and their ISs.

**Developing methodologies to make efficient use of Commercial Off The Shelf software products in Research.** Use of Commercial-Off-The-Shelf software in research is increasing as it is in the Industry. A collection of constantly updated approaches exists to provide Industry with guidelines and models on how to select, configure and use software tools, based on system requirements. The situation in Research is quite different since requirements may be constantly changing, the lifetime of the software suites tends to be short and the development effort available rather limited. The research has identified all the issues and approaches most likely to prove promising (Collignon and Cook, 2004). Work continues on case studies to assess the efficacy of the preferred approaches.

**How the audio interface for short range infrared guided air-to-air missiles can be improved.** This research undertakes a study on how the audio interface for short range infrared guided air-to-air missiles can be improved. Older analogue weapons systems allow aircrew to "listen in" on the missile's signal processing of the scene and target. This allows a great deal of information to be gained about the missile condition and its tracking performance before committing to a missile launch. The goal of this research is to extend this concept to newer missiles with all digital signal processing.

**Measures of Effectiveness: The Standards for Success.** Measures of Effectiveness (MOEs) are an important part of the process that develops a need into an acceptable solution. MOEs provide a rigorous basis for selecting between alternative solutions and a means of knowing if success has been achieved. MOEs are of importance whenever there is a need to answer the question 'How do I know if I have succeeded?' They are as applicable to simple every-day situations such as making a super-market purchase as they are to the most complex and complicated of situations such as developing an inter-planetary probe. MOEs are an intuitive human process applicable to all situations from the most trivial to the most significant situations. However, their importance does not appear to be universally recognised or appreciated. This is seen not only in the confusion caused by differences between the definitions used by different bodies but also in the lack of any literature on their theoretical basis. The confusion is compounded by the existence of a variety of terms to describe the process and by the use of these very same terms to describe other processes. There is a real need for a universally agreed definition of MOEs; for agreement on the roles of MOEs; and for the development of the theory of MOEs. This qualitative thesis examines the work of practitioners and establishes common elements of agreement relating to the definitions of MOEs. A number of research questions are posed and data is collected to provide answers to these research questions. A generic definition of the term 'Measure of Effectiveness' can be established from this data as well as by a study of the various possible meanings of the individual words making up the term. This definition provides the basis for the theory underpinning the use of MOEs. The relationship between MOEs and Measures of

Performance (MOPs) is studied in order to distinguish between these two important artefacts which are often confused with each other. This relationship is the key to the development of an elegant definition and to the establishment of a body of theory pertinent to MOEs. Following this work, guidelines for the formulation of MOEs were established and the entire process is demonstrated through the use of a case study. The case study also provides the means for determining the cognitive process underpinning the formulation of MOEs. This project shows that MOEs are the responsibility of the stakeholder and that MOEs relate to the need, but that the presence of more than one stakeholder can present a range of needs and hence a range of MOEs for the same problem. Identification of stakeholders and their domains is a prelude to the establishment of MOEs. Until there is an expressed need, there cannot be a MOE. The role of MOEs is established and in particular their role in establishing which solution best answers the needs of the stakeholder. The project demonstrates the role of MOEs as being the critical factor in establishing the objectives of the Test and Evaluation (T&E) process. A definition of MOEs is established which incorporates the common elements of the definitions of MOEs proposed by the major practitioners. A clear distinction is made between MOEs and MOPs clarifying a common mistake made by users of these terms where one is often confused for the other. The body of theory developed will enable MOEs to be used correctly and to advantage as well as being a resource for those involved in teaching the subject. Examples of MOEs are provided from various disciplines. The study concludes with the development of a heuristic process for the formulation of MOEs able to be executed in the framework of a consensus problem-solving model enabling all stakeholders to be involved in the process.

**Strategic Planning and Capability Engineering.** Systems engineering is seen as the natural methodology for the development of large, technical projects and it has received considerable intellectual effort over the years to evolve into its current state of maturity. The process that are used both in Defence and industry to determine what projects to undertake are far less well developed. This research program combines a number of tasks that are examining strategic planning methodologies and architecture-based planning approaches to identify promising approaches for structured, traceable, and defensible enterprise development decision making. (Hodge and Walpole, 1999; Staker, 2002; Cropley et al, 2003)

## **Conclusions**

A research program into systems engineering covers a broad range of activities that range from advancing the theoretical understanding of what systems engineering is and the philosophy that is adopted by its practitioners to ensure successful programs to specific areas of practice.

The incorporation of process elements into the requirement object has shown to be a promising approach for increasing the effectiveness of the systems engineering process and providing customers with a product that meets their needs.

The RISC approach to solving the problem of poor requirements is the time-phased parallel process of building a suite of tools for performing specific tasks. This is similar to the standard systems engineering process of partitioning the desired functionality of the system into subsystems and then constructing and integrating each subsystem into the desired system. Once each tool is in use, new applications and modification are suggested. The suite then evolves until much if not the entire problem is solved. While the RISC approach may not be applicable to solving all complex problems, in these applications the results have been promising. FRED, CREAP, TIGER, the remaining PETS and the OCH provide capability that shows that parts of the problem of poor requirements can be alleviated by applying the appropriate technology.

There is a wide area of research currently under way at SEEC, providing scope for research into both the breadth and depth of systems engineering as well as the theoretical and applied domains.

## References

- Carson, R.S., "Keeping the Focus During Requirements Analysis", *Proceedings of the 11th International Symposium of the International Council on Systems Engineering (INCOSE)*, Melbourne, Australia, 2001.
- Checkland P., *Systems thinking, systems practice*, Wiley, Chichester.
- Checkland, P., and Holwell S., (1998) *Information, Systems and Information Systems*, Wiley, Chichester, ISBN 0-471-95820-4.
- Cook S.C. (1990), "Knowledge-Based Generation of Measuring Instrument Specifications", in IMEKO International Symposium on Knowledge-Based Measurement, 1st, Karlsruhe, FDR, pp145-152, 1990.
- Cook S.C. (1991), A Knowledge-Based System for Computer-Aided Generation of Measuring Instrument Specifications, Ph.D. thesis, City University, London, UK.
- Cook S.C. (1993), "A Knowledge-Based Specification Generation System for Computer-Aided Production of Measuring Instrument Specifications", *Measurement*, Vol 11, pp 235-255.
- Collignon S., & Cook S., "Complex Systems, COTS and Knowledge", *Proceedings of the Systems Engineering Test and Evaluation (SETE) 2003 Conference*, Canberra 2003.
- Cook S.C, Kasser J.E., Ferris T.L J., , Elements of a Framework for the Engineering of Complex Systems, *Proceedings of the 9th ANZSYS Conference*, Melbourne, Australia, 18 - 20 November 2003
- Cook S. C., Kasser J.E., Asenstorfer J., "A Frame-Based Approach to Requirements Engineering", *Proceedings of the 11<sup>th</sup> International Symposium of the INCOSE*, Melbourne, Australia, 2001.
- Cropley, D., Yue, Y. and Cook S., "On Identifying a Methodology for Land C2 Architecture Development", *Proc of the Land Warfare Conference 2003*, Adelaide, 2003.
- Crosby, P.B. 1979, *Quality is Free*, McGraw-Hill.
- Deming, W.E. 1986, *Out of the Crisis*, MIT Center for Advanced Engineering Study., p11
- Fabryky, W. (2003), International SE Programs, presented to the INCOSE Education and Research Working Group Meeting, Washington, USA, 29 June 2003.
- Ferris T.J.L., Cook S. C. & Honour E, "A Framework for Systems Engineering", *Proceedings of the SETE 2003 Conference*, Canberra 2003.
- Gabb, A., "Front-end Operational Concepts - Starting from the Top", *The 11th Annual Symposium of the INCOSE*, Melbourne Australia, 2001.
- Glover, S.J., Bennett, K.H. "An Agent-Based Approach to Rapid Software Evolution Based on a Domain Model", *Proceedings of the International Conference on Software Maintenance*, page(s): 228 -237, 1996.
- Hodge, R. and Walpole, G. 1999, "A Systems Approach to Defence Capability Planning – A Work in Progress", in *Proceedings and on CD-ROM of the SETE99 Systems Engineering Test and Evaluation Conference, Conceiving, Producing and Guaranteeing Quality Systems*, Adelaide, Australia, 20-22 October 1999, pp. 21-32.
- Hooks, I., "Writing Good Requirements", *Proceedings of the 3rd NCOSE International Symposium, 1993*, available at <http://www.incose.org/rwg/writing.html>, last accessed November 1, 2001.

- Hopkins, F.W., Rhoads, R.P. "Object Oriented Systems Engineering – An Approach", *Proceedings of the 8th International INCOSE Symposium*, 1998.
- Jacobs, S., "Introducing Measurable Quality Requirements: A Case Study", *IEEE International Symposium on Requirements Engineering*, Limerick, Ireland, 1999.
- Kasser J.E., "The Cataract Methodology for Systems and Software Acquisition", *Proceedings of the SETE 2002 Conference*, Sydney Australia, October 2002.
- Kasser J.E., "Does Object-Oriented System Engineering Eliminate the Need for Requirements?", *Proceedings of the 13<sup>th</sup> International Symposium of the INCOSE*, Washington DC., 2002a.
- Kasser J.E., "Object-Oriented Requirements Engineering and Management", *Proceedings of the SETE 2003 Conference*, Canberra, 2003.
- Kasser J.E. "A Prototype Tool for Improving the Wording of Requirements", *Proceedings of the 12<sup>th</sup> International Symposium of the INCOSE*, Las Vegas, NV, 2002b.
- Kasser J.E., *Applying Total Quality Management to Systems Engineering*, Artech House, June 1995
- Kasser J.E., "ELMER: An Expert System Based on a Finite State Machine", *Proceedings of the AMSAT Symposium*, Washington DC., 1992.
- Kasser J.E. and Cook S.C. "The Communications Requirements Evaluation & Assessment Prototype (CREAP)", *Proceedings of the 12<sup>th</sup> INCOSE*, Las Vegas, NV, 2002.
- Kasser, J.E. and Massie A., "A Framework for a Systems Engineering Body of Knowledge", *Proceedings of the 11th Annual International Symposium of the INCOSE*, Melbourne, Australia, 1-5 July 2001, Paper No. 8.5.3.
- Kasser, J.E., Schermerhorn R., "Determining Metrics for Systems Engineering", *Proceedings of the 4th International Symposium of the INCOSE*, San Jose, CA., 1994.
- Kasser J.E., Cook S.C.C., "Using a Rapid Incremental Solution Construction Approach to Maximise the Completeness and Correctness of a Set of Requirements for a System", *Proceedings of the 13<sup>th</sup> International Symposium of the INCOSE*, Washington DC., 2000.
- Kasser J.E., Cook S.C., Scott W, Clothier J., Chen P., "Introducing a Next Generation Computer Enhanced Systems Engineering Tool: The Operations Concept Harbinger", *Proceedings of the SETE 2002 Conference*, Sydney Australia, October 2002.
- Kasser J.E., Shoshany S., "Systems Engineers are from Mars, Software Engineers are from Venus", *The 13th International Conference on Software & Systems Engineering and their Applications*, Paris, France, 2000 .
- Kasser J.E., Tran X-L, Matison S., "Prototype Educational Tools for Systems and Software (PETS) Engineering", *Proceedings of the 14<sup>th</sup> Annual Conference for Australian Engineering Education*, Melbourne, Sept-Oct 2003.
- Kuhn, T.S., 1970, *The Structure of Scientific Revolutions*, The University of Chicago Press, Second Edition, Enlarged.
- Kline S.J., *Conceptual Foundations for Multidisciplinary Thinking*, Stanford University Press, California, ISBN 0-8407-2409-1, 1995.
- Lander S.E., *Issues in Multiagent Design Systems*, IEEE Expert, Volume: 12 Issue: 2, March-April 1997.
- Leibrandt, R. What is the INCOSE guide to the systems engineering body of knowledge (SEBOK)?, *Proceedings of the 11th Annual International Symposium of the INCOSE*, Melbourne, Australia, 1-5 July 2001, Paper No. 8.5.1.
- Lykins, H., Friedenthal, S., Meilich, A., "Adapting UML for an Object Oriented Systems Engineering Method (OOSEM)", *Proceedings of the 10th International INCOSE Symposium*,

2000.

Meilich, A., Rickels, M., "An Application of Object Oriented Systems Engineering (OOSE) To an Army Command and Control System: A New Approach to Integration of System and Software Requirements and Design", *Proceedings of the 9th International INCOSE Symposium*, 1999.

*NASA Systems Engineering Handbook*, draft, September 1992.

Nii H.P., "Blackboard Systems", Knowledge Systems Laboratory Report No. KSL 86-18, Knowledge Systems Laboratory, Department of Medical and Computer Science, Stanford University, 1986.

Schach S., *Object-Oriented and Classical Software Engineering*, p 294, McGraw Hill, 2002.

Staker, R. 2002, "Bayesian Decision-Theoretic Planning for Complex Systems-of-Systems", IEEE Information, *Decision and Control Conference, Adelaide, Australia*, 11 to 13 February 2002, pp 413-418.

Standish (1995), Chaos, The Standish Group, <http://www.standishgroup.com/chaos.html>, last accessed March 19, 1998.

Sutcliffe, A., Galliers, J., Minocha, S., "Human Errors and System Requirements", *IEEE International Symposium on Requirements Engineering*, Limerick, Ireland, 1999.

Van Gaasbeek, J. R., Martin, J. N., "Getting to Requirements: The W5H Challenge", *The 11<sup>th</sup> Annual Symposium of the INCOSE*, Melbourne Australia, 2001.

Voyages (1996), "Unfinished Voyages, A follow up to the CHAOS Report", The Standish Group, [http://www.pm2go.com/sample\\_research/unfinished\\_voyages\\_1.asp](http://www.pm2go.com/sample_research/unfinished_voyages_1.asp), last accessed January 21, 2002.

## Authors

**Joseph Kasser** has been a practising systems engineer for 30 years. He is the author of "*Applying Total Quality Management to Systems Engineering*" published by Artech House. Dr. Kasser holds a Doctor of Science in Engineering Management from The George Washington University, and is a Certified Manager. Currently, he is the Defence Science and Technology Organisation Associate Research Professor at the Systems Engineering and Evaluation Centre at the University of South Australia (UniSA). He performs research into improving the Defence acquisition process, the nature of systems engineering and the properties of object-oriented requirements. He is also the Project Manager for the Defence Materiel Organisation customised section of the Master of Project Management postgraduate degree. Prior to taking up his position at UniSA, he was a Director of Information and Technical Studies at the Graduate School of Management and Technology at University of Maryland University College. There, he developed and was responsible for the Master of Software Engineering degree and the Software Development Management track of the Master of Science in Computer Systems Management (CSMN) degree. He is a recipient of NASA's Manned Space Flight Awareness Award for quality and technical excellence (Silver Snoopy), for performing and directing systems engineering. Dr. Kasser also teaches systems and software engineering in the classroom and via distance education.

**Professor Stephen Cook.** After graduating in Electronics Engineering from the South Australian Institute of Technology in 1977, Prof Cook commenced work as a telecommunications equipment design engineer in the UK where he also completed an MSc in Computer Science at the University of Kent. On his return to Australia, he worked in the defence electronics industry

until 1988 when he joined the Defence Science and Technology Organisation (DSTO). He completed a PhD in 1990, in Measurement Science and Systems Engineering at the City University London. Prof Cook joined the University of South Australia as the Foundation Professor of Systems Engineering in 1997. He has contributed to three books and has published over fifty refereed journal and conference papers. Prof Cook is a Fellow of the Institution of Electrical Engineers (UK), a Fellow of the Institution of Engineers, Australia and Past President of the Systems Engineering Society of Australia.