Introducing a Next Generation Computer Enhanced Systems Engineering Tool: The Operations Concept Harbinger

Joseph E. Kasser DSc CEng CM MIEE
Stephen C. Cook PhD FIEE FIEAust
William Scott
Systems Engineering and Evaluation Centre
University of South Australia
School of Electrical and Information Engineering F2-37
Mawson Lakes Campus
South Australia 5095
Telephone: +61 (08) 8302 3941
Fax: +61 (08) 8302 4723
Email: joseph.kasser@unisa.edu.au

Jennie Clothier PhD
Pin Chen PhD
Joint Systems Branch
Defence Science & Technology Organisation
Department of Defence
Fernhill Park
Canberra
ACT 2600, Australia
Tel: 61-2-62566181
Fax: 61-2-62566130
E-mail: pin.chen@dsto.defence.gov.au

ABSTRACT

This paper proposes a new generation Computer Enhanced Systems Engineering (CESE) tool known as an Operations Concept Harbinger (OCH). The OCH is the result of the application of information management technology to solving the problem of poorly implemented requirements engineering, in particular poorly written and articulated requirements as well as the effect of changing requirements for both single systems and Systems of Systems. The OCH thus bridges the gap between the soft systems methodologies used in the early phases of the system development life cycle 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 paper also describes the experiences of prototyping the OCH for a Force Level Systems Engineering application.

INTRODUCTION

We are living in an age of transitions. One of these is the transition from hardware-based systems to software-based systems. While Information Technology (IT) underpins the civilization in the early years of the 21st Century, the current systems and software acquisition and development paradigm is characterized by mind-boggling complexity, a turbulent transition from function-based designs to object-oriented designs, cost and schedule overruns (CHAOS 1995; OASIG 1996), and project failures (Capers Jones 1996).

Hooks (1993) discussed the problem of poorly written requirements. Kasser and Williams (1998) show that a major cause of these cost and schedule overruns is the result of poor requirements engineering, namely

- Missing and incorrect requirements;
- Poorly written and articulated requirements;
- Failure to manage changing requirements during the course of the construction of the system.

Moreover, Jacobs (1999) states that a 1997 analysis of the software development process performed at Ericsson identified *"Missing Understanding of Customer Needs*" as the main obstacle for decreasing fault density and lead-time. Moreover, during the years since the earlier papers were published, requirements management tools have proliferated, but they do not seem to be addressing problem of poorly written requirements. Thus, Carson (2001) addresses these same issues years after Hooks (1993) raised them. Hence the application of tool technology to the subject of reducing the effect of poorly written requirements is long overdue.

The problems experienced in creating an optimal single system on schedule and within budget are minor when compared with the problems of coping with the acquisition and life-cycle management of multiple systems. Recognition of this expansion of the scope of the problem to deal with multiple systems at different stages of evolution has resulted in the modern emphasis on what has become known as "Systems of Systems". Moreover, increasing complexity of the evolution of Systems of Systems and their systems engineering context is challenging many aspect s of traditional systems engineering practice (Chen and Clothier 2002).

This paper introduces a new Computer Enhanced Systems Engineering (CESE[1]) tool known as an Operations Concept Harbinger (OCH). This tool 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 (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.

CONCEPTS OF OPERATIONS

Operations concept documents (OCD) are one of the critical documents in the systems development lifecycle (Kasser and Schermerhorn, 1994) because they provide the vision of the proposed system. Gabb (2001) summarizes the purpose of the OCD as describing the operation of a system in the terminology of its users. Gabb stated that it may include identification and discussion of the following

- Why the system is needed and an overview of the system itself.
- The full system life cycle from deployment through disposal.
- Different aspects of system use including operations, maintenance, support and disposal.
- The different classes of user, including operators, maintainers, supporters, and their skills and limitations.
- Other important stakeholders in the system.
- The environments in which the system is used and supported.
- The boundaries of the system and its interfaces and relationships with other systems and its environment.
- When the system will be used, and under what circumstances.
- How and how well the needed capability is currently being met (typically by existing systems).
- How the system will be used, including operations, maintenance and support.

Gabb also provides the traditional system engineering perspective when he writes that "an OCD is not a specification or a statement of requirement - it is an expression of how the proposed system will or might be used, and factors which affect that use. As such it is not obliged to follow the 'rules' of specification writing and can be relatively free in its language and format. Generally it will contain no 'shalls'." However, a true "free form" OCD tends to complicate traceability of subsequently written requirements to sections of the OCD.

THE REQUIREMENTS PROCESS

Requirements management. In the course of the System Development Life Cycle, the OCD is analysed and used to create the Requirements Document. However, it is well known that contemporary requirements management practice is far from ideal. This inadequacy shows up in various ways including

- **Vague and unverifiable requirements** – due to poor phrasing of the written text.
- **Incompletely articulated require-**

[1] The term CESE was introduced in 1996 to distinguish between computer tools for systems engineering and the commonly used acronym CASE meaning Computer Aided Software Engineering.

ments – due to a poor requirements elicitation process.

- **Incomplete requirements** – due to various factors including domain inexperience, and the lack of expertise in eliciting requirements by technical staff.
- **Poor management of the effect of changing user needs during the time that the system is under construction** – due to lack of the understanding of the need for change management, and the tools to do the function in an effective manner.

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, "we don't perform system engineering to get requirements and, "we perform system engineering to get systems that meet specific needs and expectations." The focus is on user needs, not requirements. What system engineers appear to have forgotten is that requirements are used to document user needs in a verifiable manner, requirements are a means, not an end. There is nothing divine about requirements they are just a translation of user needs into function and performance specifications for the system to be built.

Change is pervasive. User needs are not static, they change over time, and it must be expected that they will change while the system is being constructed. The software world has recognized this situation and has largely abandoned the waterfall methodology in favour of a variation of the spiral and rapid development methodologies. The objected-oriented software world has gone one step further and eliminated the use of requirements by representing the user needs in the form of *use cases* and Unified Modelling Language (UML) diagrams. Schmuller (1999) states that UML diagrams may be enhanced by using pictures to make them more understandable by the non-technical viewer. The UML diagrams provide a number of temporal and relational views of a system.

The UML *use case* scenarios are very similar to the scenarios in the traditional systems engineering concepts of operations so there is nothing to preclude the use of *use cases* for describing all of Gabb's points. While the object-oriented approach uses the UML as a convenient way of expressing use cases, and improves the representation of user needs, it has not provided much to improve the elicitation of the user needs.

Poor reliability. The current process of documenting and managing user needs includes two sequential steps, namely first producing an OCD and then a System Requirements Document (SRD). Each of these processes is imperfect and introduces errors into the products. These errors are then carried through into the design and implementation of the system.

UML diagrams captured in a database type of tool may be used to describe scenarios and replace OCDs. Current thought in Requirements Engineering is to consider the SRD as a printout of a System Requirements Database. Yet while traceability of requirements to the OCD is advocated, the concept of combining or connecting the UML OCD and the System Requirements database does not seem to have been explicitly stated. It is however implied by the object-oriented approach of modelling that commences construction of a system based on the UML diagrams. Holt (2001 p33) states that many projects succeed due to effective modelling (the 'M' in UML). The OCH makes use of these concepts.

A NEW TYPE OF REQUIREMENTS MANAGEMENT TOOL

Tools 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 case*. 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.

Current UML descriptions are static representations. Schmuller (1999) stated that the UML diagrams can be enhanced by using graphic depictions of components instead of boxes to enhance the communication. So why not further enhance the *use case* representation in the tool by the addition of multimedia descriptions and other representation diagrams. Thus the *use cases* would be presentable in the form of a mixture of various formats, as appropriate, including but not limited to

- UML, basic and enhanced with graphics – the current way of working.
- Static workflow diagrams – adding user perspective, the tool would transform the diagrams into UML for the developers and testers.
- Dynamic workflow diagrams – animated diagrams showing sequences.
- Audio files – containing verbal descriptions such as environmental conditions for the deployed system, or user concerns.
- Video files – showing scenarios. These may be animated or filmed.
- Simulations – of all or part of the system.
- Text mode – for those people who still think in terms of “requirements”.

By presenting the concept of operations of the proposed system in this type of tool, the description should be more comprehensive than that in purely text mode OCDs.

The tool database is then further expanded into an integrated project technical and managerial database by adding the Quality System Elements (QSE) (Kasser 2000). These data are present in the existing production paradigm, but tend to be separated in several different tools: Requirements Management, Project Management, WBS, Configuration Control, and Cost Estimation, etc. The QSE in the OCH facilitate the next step in the evolution of Requirements Engineering; a discipline that is evolving from its traditional role as a mere front-end to the systems life cycle towards a central focus of change management in system-intensive organizations (Jarke 1996). Kasser not only describes the expansion of the traditional Requirements Traceability Matrix into an integrated technical and programmatic project database containing the QSE, he also describes the concept of tools and agents that access the QSE to provide various views into the state of the project. The OCH is such a tool

THE OPERATIONS CONCEPT HARBINGER

The OCH bridges the gap between the soft systems methodologies used in the early phases of the system development life cycle and the hard systems methodologies used in the construction of a system. The OCH is

- A new type of CESE tool.
- A tool for capturing user needs without the explicit use of “text-mode requirements”.
- Optimal for systems that span 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.

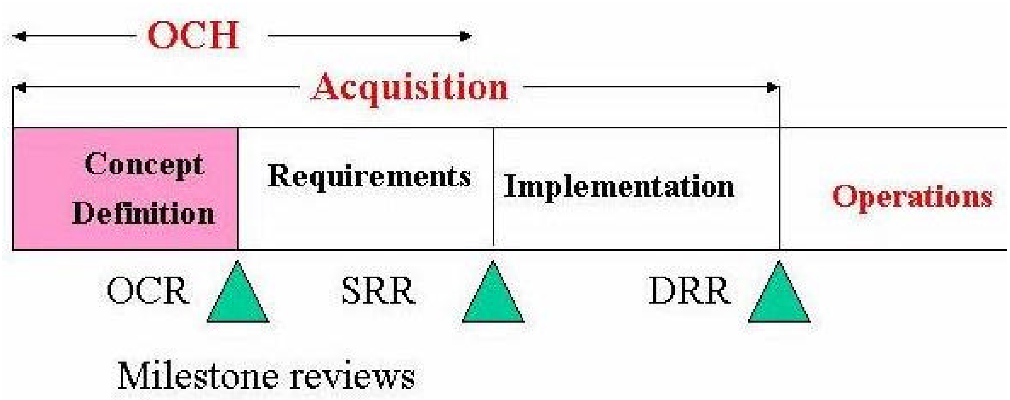

Figure 1 The Place of the OCH in the System Life Cycle

- Scenario based.
- Diagram or picture based.

Figure 1 shows 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, and full-scale simulations. 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.

DESCRIPTION OF THE OCH

The internal structure of the OCH is frame -based (Cook et al., 2001) where each frame may be considered as a page of information. The contents of the frame (page) controls which information is to be displayed on the computer display or on a large multimedia screen display. The frame has two dimensions as shown in Figure 2. The vertical (temporal) dimension controls the sequence and duration of each display page, and the horizontal (activity) dimension contains the individual screen display information showing the element of the use case that is active in the specific time frame for all parts of the system. This is the same format as the UML swim lane activity diagram. The difference being that each of the blocks can be animated, and the frame also has slots for the MOE tags and other QSE elements for each activity in the *use case*.

Form of the OCH. The OCH takes the form of a set of display pages or screens within the two-dimensional structure as shown in Figure 3. Each display page can contain a number of different items as shown in Figure 4, namely

- A multimedia multi-screen presentation containing multi-media as appropriate, process flow-charts, video of actual or similar scenarios. This may be thought of as being an animated UML swim lane diagram.
- A storyboard screen for continuity.

Multiple screens as appropriate, e.g. depth - drilling down to details as appropriate.

-

Non-linear access to operations concept. Each page contains hyperlinks to other pages. This feature allows for quick access to various views and sections of the system: a feature that is not possible with a paper-based OCD.

In a multiple system situation. When scenarios cross systems boundaries, each screen is a representation of the scenario in the system allocated to the screen. There are two main *use cases* for the OCH in the front-end of the system life cycle, as

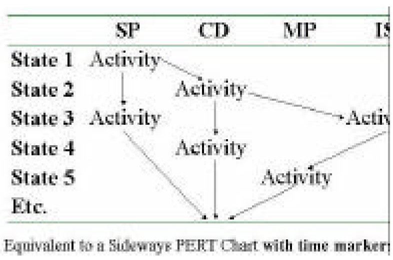

Figure 2 Two-dimensional format of OCH internal structure

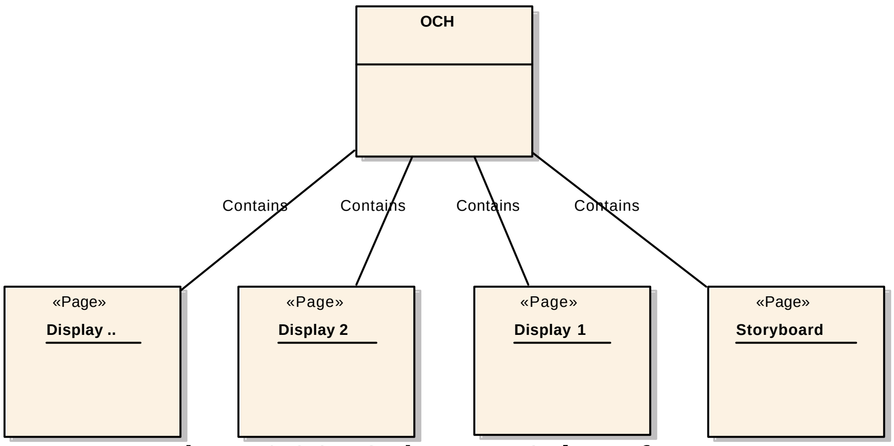

Figure 3 OCH Swim Lane Column format

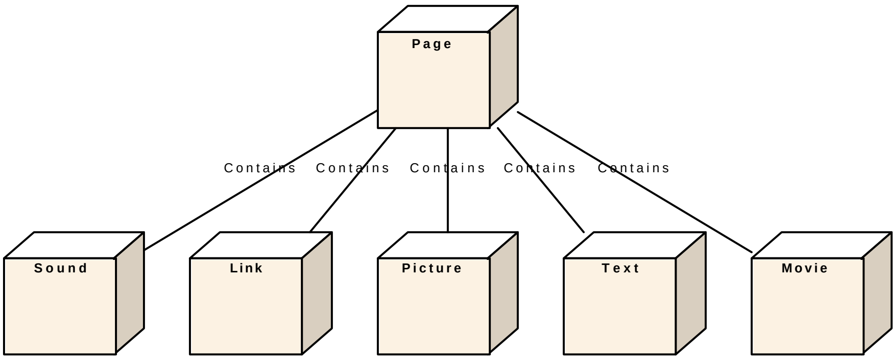

Figure 4 OCH Display Page Contents

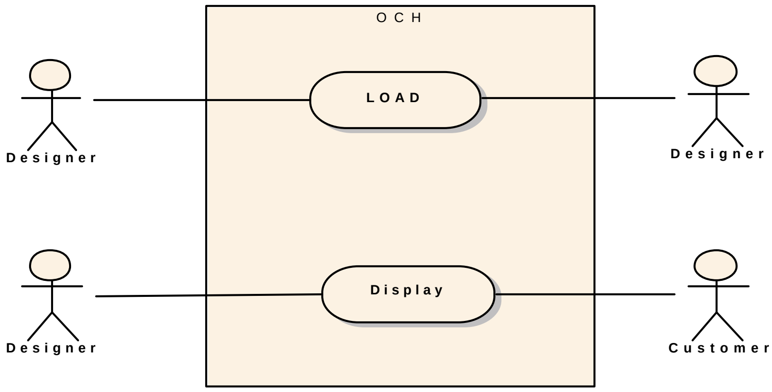

Figure 5 OCH Use Cases

shown in Figure 5. These are

- Loading the *use cases* into the OCH, and
- Displaying and discussing the contents of the OCH.

ENHANCING THE REQUIREMENTS ELICITATION PROCESS.

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 the soft systems methodologies such as the CATWOE (Checkland 1993) approach 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*. This process, depicted in Figure 6 is an enhanced requirements elicitation process. The interactive visual approach improves the communications between the system engineer and the customer in documenting the user needs.

ADVANTAGES OF THE OPERATIONS CONCEPT HARBINGER

The advantages of the OCH include but are not limited to the following

- Bypassing today's problems due to "poor requirements".
- Visual representation of "needs" leads to a dialogue based on the following elicitation questions
    1. **Is this what you want?** Which tends to clarify misconceptions between user and analyst as well as increasing the articulation of needs.
    2. **On what basis will you decide that your need is met by the system in this scenario?** Which leads to the MOEs and QSE.
- Visual representation readily leads to inherited "needs" which reduces the quantity of "missing requirements". For example if the visual representation shows a desert environment, the discussion should also focus on environmental and regulatory needs, which can be, and have often been, forgotten in today's paradigm.
- Facilitates shared meaning by virtue of being able to view the contents in a non-sequential manner. This allows the conversations to rapidly move to other parts of the system by virtue of a few clicks on hyperlinks.
- Reduces reliance on words. This is becoming increasingly important in today's climate of global alliances for major systems development.
- Can show more than one view of a system at the same time. These views may be spacial or temporal. For example, in a business process-reengineering situation, an "as-is" version can be displayed together with a "to-be" version of a system and the differences can be clearly articulated.

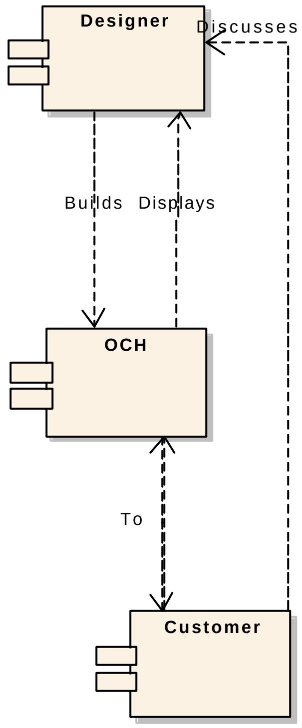

Figure 6 Enhanced Needs Elicitation Process

DISADVANTAGES OF THE OPERATIONS CONCEPT HARBINGER

The disadvantages of the OCH are

- It is easy to state "needs" in terms of "solutions" since representations tend to be in the form of examples of solutions. For example a video showing a proposed system in operation will tend to show how the system will work as well as what it will do. However, this disadvantage may be put to use in a verbal proposal situation, in which the OCH may be used to describe how a proposed system will meet the purchaser's needs. The verbal proposal can be in the form of a multimedia OCH that can be viewed at the purchaser's convenience and remains on record for the duration of the project.
- It is more difficult to construct than a written document. However, this may be an artefact of the process.

FORCE LEVEL SYSTEMS ENGINEERING

Force Level Systems Engineering (FLSE) is the application of systems engineering at the Force Level that is to multiple systems. The Australian approach to FLSE is different to that of the United States Navy (USN) Battle Force Systems Engineering (BFSE) (Dickerson and Soules 2002). However, the Australian Defence Force (ADF) does not yet have the USN like Integrated Digital Environment (IDE) to support the process. Consequently, the recommended Australian approach is to first review the concepts developed in the USN and other countries, and then adapt them for the ADF in such a manner as to evolve the current FLSE into the ADF IDE (AIDE).

In early 2002, the Joint Systems Branch (JSB) of the Defence Science and Technology Organisation (DSTO) was carrying out a number of initiatives in architecture and systems engineering for joint C4ISREW systems analysis, future capability studies and force development.

The teams working in these initiatives produced many interesting and important findings, useful data and references for supporting force level defence capability planning and management, and gained good experience in using various tools for architecture development and analysis. JSB was then faced with the problem of how to bring theses outcomes together and indicate the feasibility of FLSE across the strategic and capability disciplines in a relatively simple demonstration.

THE FLASH

The dimensions of the problem include

- Process and process interactions.
- Tools required at each level.
- Communications and information flows.
- Data, change and configuration management.
- Organizational cultural imperatives.

The solution was to present the outcomes in the form of a Force-Level ADF Systems Harbinger (FLASH) based on the OCH concept.

The purpose of the FLASH is to articulate the concept of an integrated toolset that maps onto existing FLSE processes.

The value of FLASH is to

- Illustrate how decision making at all levels can be improved through the provision of timely, correct and concise information.
- Identify and refine processes in all areas.
- Identify and refine process products.
- Identify and improve information flows and interfaces; in particular between areas.
- Identify possibilities for automated analysis.Identify how timely impact assessments can be facilitated.

The scope of the FLASH. The scope of the FLASH is a framework for an engineering environment encompassing four areas

- Strategic Planning;
- Capability Development;
- In-service capabilities;
- Major projects within any phase in their lifecycle.

BUILDING THE FLASH

The activities in Capability Development were analysed using the CATWOE (Checkland 1993) approach. Relevant documentation was obtained and reviewed, and cognizant personnel were interviewed. The result of the analysis was documented in the OCH. Other pertinent information was recorded as appropriate in UML style diagrams.

The initial structure of the FLASH was based upon the IEEE P1362 template for an OCD to provide some degree of familiarity and compatibility with existing methodologies. However, it was found that the one-dimensional sequential layout of the OCD did not transfer well to a multimedia non-sequential presentation format. In particular, the separation of current practices from future scenarios by the description of the proposed system created a gap preventing direct comparison of the "as-is" with the desired "to-be" descriptions of the Capability Development situation.

The Mark 1 PowerPoint-based FLASH. The concept of multiple screens of pages of information viewed in parallel was met in the Mark 1 implementation by a series of up to five PowerPoint presentations displayed on up to five computers and viewed simultaneously. The PowerPoint presentations were scripted to present the operations concept. Each PowerPoint slide could contain text (dot points), sounds, graphics, and moving pictures. This was standard PowerPoint technology and the set of linked OCH presentations was easy to construct by ensuring that each presentation contained the same number of slides and the slide numbers corresponded to each other in each presentation. Each set was dedicated to a part of the modelled process. While effective in conveying the desired information, the presentations were essentially individual with no direct association. Difficulty was found in the synchronization of the presentations. The difficulty was in linking the five presentations so that motion of one mouse was repeated on each presentation. This was a desired capability to enable a single person to control all presentations. The problem was initially solved by the use of a set of radio-controlled mice on a single frequency, which allowed all of the computers to receive and react to a signal from a single transmitting mouse. While the hardware solution worked, it required non-standard hardware and as such was considered less than optimal since there was a need to make the OCH portable. As a consequence, while the OCH concept had been demonstrated, a software solution was deemed necessary for a practical tool.

The Mark 2 Web-based FLASH. The software approach for synchronized displays was based on the use of Web enabled technology in a client-server relationship. The server was equipped with a controlling script and sets of pages for display. When queried by the control client, the server returned the appropriate page for display. The server kept track of the page being displayed at each client. This allowed customisable transitions in the presentation to accommodate the audience with all the flexibility provided by web technology. The use of standard web pages allowed the construction of the pages using simple HTML editors and a skeleton script was written for the server and then customised to each page of the presentation.

The web pages are also linked allowing construction of demonstrators that may pass information from one view to another. This was unachievable using PowerPoint and provides the web-based solution with a 'playability' advantage. The drawback at this time is all page transitions need to be manually linked. This means that complex presentations are more difficult to construct than simple presentations. However, there is no intrinsic reason why the next iteration of the OCH should not contain features that facilitate the construction of links.

The FLASH OCH. The FLASH OCH is in an early stage of its evolution. At this time, it contains

- Examples of the USN approach to BFSE for reference and comparison purposes.
- A representation of the current ADF capability planning process.
- Examples of a proposed future ADF capability planning process within a future, yet to be developed, AIDE.

Early Results. The OCH approach to solving the problem of presenting a relatively simple demonstration of the capability of FLSE has shown itself to be both practical and useful. The ability to examine a system in a non-sequential manner via the built-in hyperlinks within the tool has provided the capability to view the contents of the FLASH from various perspectives. The discussions around the multimedia contents of the frames have facilitated the needs elicitation process by showing the effects of various approaches and have lead to the identification of missing needs.

Research into integrated project databases, the OCH, and the FLASH implementation continues.

SUMMARY

This paper has described the OCH, the first of a proposed new generation of CESE tools based on an integrated project database architecture. The OCH has the potential to minimize the cost and schedule overruns currently experienced in systems and software development by bypassing the problems inherent in poor requirements engineering.

CONCLUSION

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 eliciation and identifaction process is enhanced by the multimedia technology embodied in the OCH.

REFERENCES

Capers Jones, *Patterns of Software Systems Failure and Success*, International Thomson Computer Press, 1996.

Carson, "Keeping the Focus During Requirements Analysis", *The 11th INCOSE International Symposium, Melbourne, Australia, 2001*.

CHAOS, "The CHAOS Report", The Standish Group, 1995, http://www.pm2go.com/sample_research/chaos_1994_1.asp, last accessed January 21, 2002.

Checkland, P. Systems Thinking, Systems Practice, 1993, pp. 224-225.

Chen, P., Clothier, J., "Advancing Systems Engineering for Systems-Of-Systems Challenges", *The 12th Annual Symposium of the INCOSE*, Las Vegas, NV, 2002.

Cook S. C., Kasser J.E., Asenstorfer J., "A Frame-Based Approach to Requirements Engineering", *The 11th International Symposium of the INCOSE*, Melbourne, Australia, July 2001.

Dickerson, C., Soules, S., "Using Architecture Analysis for Mission Capability Acquisition", Proceedings of the ICRTS, April 2002, available at http://www.dodccrp.org/Activities/Symposia/2002CCRTS/Proceedings/Tracks/pdf/123.PDF, last accessed September 12, 2002.Gabb, A., "Front-end Operational Concepts - Starting from the Top", *The 11th Annual Symposium of the INCOSE*, Melbourne Australia, 2001.

Gabb, A., Caple, G., Haines, D., Lam ont, D., Davies, P., Hall, A., van Gaasbeek, J., Eppig, S., Jones, D., Vietinghoff, B., "Requirements Categorization", *The 11th Annual Symposium of the INCOSE*, Melbourne Australia, 2001.

Hooks, I., "Writing Good Requirements", *Proceedings of the 3rd NCOSE International Symposium, 1993*, http://www.incose.org/rwg/writing.html, last accessed November 1, 2001.Jacobs, S., "Introducing Measurable Quality Requirements: A Case Study", *IEEE International Symposium on Requirements Engineering*, Limerick, Ireland, 1999.

Jarke, M., Meta Models for Requirements Engineering, *Tenth Knowledge Acquisition for Knowledge-Based Systems Workshop*, Banf, Alberta, Canada, 1996, http://ksi.cpsc.ucalgary.ca/KAW/KAW96/jarke/Jarke.html, last accessed September 12, 2002.

Holt J., *UML for Systems Engineering watching the wheels*, The Institution of Electrical Engineers, London, 2001.

Kasser, J.E., Schermerhorn, R., "Gaining the Competitive Edge through Effective Systems Engineering", *the 4th Annual International Symposium of the National Council on Systems Engineering (NCOSE)*, San Jose, CA., 1994.

Kasser J.E., Williams V.R., "What Do You Mean You Can't Tell Me If My Project Is in Trouble?", *First European Conference on Software Metrics (FESMA 98)*, Antwerp, Belgium, 1998

Kasser J.E. "A Framework for Requirements Engineering in a Digital Integrated Environment", *the Systems Engineering Test and Evaluation Conference (SETE 2000), Brisbane, Australia, 2000*.

Kasser, J.E., "Does Object-Oriented System Engineering Eliminate the Need for Requirements?", *The 12th Annual Symposium of the INCOSE*, Las Vegas, NV, 2002.

OASIG, The performance of information technology and the role of human and organizational factors. Report to the Economic and Social Research Council, UK, 1996, available at http://www.shef.uc.uk/~iwp/publications/reports/itperf.html, last accessed January 16, 2002.

Schmuller, J. "*SAMS Teach Yourself UML* *in 24 Hours*", Sams Publishing, 1999.

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

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 is a DSTO Associate Research Professor at the University of South Australia (UniSA). He performs research into improving the acquisition process. 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.

Mr. William Scott graduated in Applied Mathematics and Computer Science at University of Adelaide in 1999, after which he was granted a research scholarship at the System Engineering and Evaluation Centre at the University of South Australia. His time spent at SEEC has been spent in the examination and implementation of the AP-233 standard.

Dr Jennie Clothier is a Senior Principal Research Scientist and Research Leader of the Joint Systems Branch, DSTO, leading research teams working in various areas related to joint defence systems including joint systems for C4ISREW, systems engineering for defence modernisation, architecture, force and capability development, and learning organisations. Dr Clothier is the Australian representative and the current Chair of the Technical Panel 4 (TP-4) (Systems Engineering for Defence Modernisation) of The Technical Cooperation Program (TTCP) Joint Systems Analysis (JSA).

Dr Pin Chen is a Senior Research Scientist in the Information Architectures Group of Joint Systems Branch, DSTO, leading research tasks on Architecture Practice Study and Systems Engineering for Systems of Systems. His current research interests include architecture practice study, software architecture, enterprise architecture, architecture methodologies, architecture supporting tools and environments, architecture capabilities integration and systems engineering processes for Systems of Systems evolution. Dr Chen joined DSTO in March 1996 after he completed his Ph.D. in Computer Science at the Australian National University.