HAL
archives-ouvertes.fr

Systems engineering research

Abd-El-Kader Sahraoui, Dennis Buede, Andrew Sage

▶ To cite this version:

Abd-El-Kader Sahraoui, Dennis Buede, Andrew Sage. Systems engineering research. Journal of Systems Science and Systems Engineering, Springer Verlag (Germany), 2008, 17 (3), pp.319 - 333. 10.1007/s11518-008-5083-9 . hal-01704863

HAL Id: hal-01704863
https://hal.laas.fr/hal-01704863

Submitted on 14 Feb 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

SYSTEMS ENGINEERING RESEARCH

Abd-El-Kader Sahraoui[1] **Dennis M. Buede**[2] **Andrew P. Sage**[3]
[1]LAAS-CNRS and Université de Toulouse
7 Avenue du Colonel Roche Toulouse 31077 France
sahraoui@laas.fr
[2]Innovative Decisions, Inc.
2139 Golf Course Dr. Reston, VA 20191 USA
dbuede@innovativedecisions.com
[3]George Mason University
Dept. of Systems Engineering and Operations Research
Fairfax, VA 22030 USA
asage@gmu.edu

Abstract
In this paper, we propose selected research topics that are believed central to progress and growth in the application of systems engineering (SE). As a professional activity, and as an intellectual activity, systems engineering has strong links to such associated disciplines as decision analysis, operation research, project management, quality management, and systems design. When focussing on systems engineering research, we should distinguish between subjects that are of systems engineering essence and others that more closely correspond to those that are more relevant for related disciplines..

1. Introduction: Systems Engineering Facets

Some key facets of systems engineering that are relevant to most research topics are the operational concept, system boundary, requirements, requirements management, functional analysis, physical and implementation architectures and interfaces, trade-off decisions, prototype development and testing, test design, and team activities. It is important that SE research agenda items enable progress in each of these facets.

1. **Operational Concept.** This represents the concept selected for system development. It is very important as mission requirements, and measures of effectiveness to which the system must contribute, follow directly from the operational concept. The operational concept also represents the flows of inputs and outputs between the system and other systems as the system interacts with other systems in fulfilling the mission.
2. **System Boundary.** This is a depiction of all inputs and outputs of a system with references to their origin or destination. An external systems diagram is generally needed to represent the system boundary.
3. **Requirements.** These represent capabilities that the system must provide. *System requirements* represent requirements about the system's inputs, outputs, external interfaces, and such system wide characteristics as cost, schedule, and reliability. *Derived requirements* represent technological requirements concerning operability and generally concern inputs, outputs, interfaces, and other technological characteristics.
4. **Requirements management**. This concerns the ability to describe and follow the life of a requirement, in both a forwards and backwards direction (Gotel and Finkelstein, 1994) throughout the lifecycle of engineering a system.
5. **Functional analysis**. This involves examination of the functions the system will have to perform across scenarios that flow from the Operational Concept. It also involves

development of a coherent functional architecture to achieve the capabilities and requirements that flow from the operational concept.

6. **Physical and implementation architectures and interfaces**. The activities here involve development of a physical representation of the system, translation of this into an architecture capable of implementation, and associated segmentation of the system into components with connecting interfaces.
7. **Trade-off decisions**. These are decision in which a choice is made among competing concerns regarding the pros and cons for each architecting and design alternative, and where one alternative is chosen as the "best".
8. **Prototype development and testing**. This involves creation and test of models of the system such that a learning process for creating design ideas and determining associated performance results.
9. **Test design**. This involves consideration of the test efforts as a system and designing the test protocols prior to implementation. *Verification* is testing to determine that the system, or a component of the system, has met its requirements. *Validation/acceptance* involves testing to demonstrate that the system meets the mission needs defined in the operational concept.
10. **Team activities**. Generally, many systems engineering activities involve the use of teams to gain synergistic capabilities across the expertise of individuals.
11. **Systems of systems (federation of systems, systems families).** This involves the taking into account the complexity of systems such evolutionary and adaptive development and emergent behaviour.

Each of these SE facets is important in systems engineering practice and research. They are major aspects studied in systems engineering case study research (Friedman and Sage, 2004).

2. RESEARCH IN SYSTEMS ENGINEERING

Honour and Buede (1998) assembled a research agenda for the International Council on Systems Engineering (INCOSE), INCOSE as part of an effort to launch the INCOSE Systems Engineering Center of Excellence (SECOE), which was intended to be a virtual network of researchers. INCOSE is a not-for-profit membership organization founded in 1990. Our mission is to advance the state of the art and practice of systems engineering in industry, academia, and government by promoting interdisciplinary, scalable approaches to produce technologically appropriate solutions that meet societal needs. . The major topic areas in this research agenda were: Value of Systems Engineering and Elements of SE; Human Productivity in SE Activities; SE Processes and Process Improvement; SE Methods; SE Automation; and Formal Methods for SE.

We expand upon this effort here by identifying and characterizing five major systems engineering research issues that support this agenda. For each research problem we will (a) define the problem being addressed, (b) identify the research approach, and (c) discuss the research results that would be expected. The research issues are: (1) agile requirement engineering in a requirement volatility context, (2) ontology for requirements elicitation, (3) a decision focused framework for life-cycle based architecting and design in SE, (4) architecting and design of the overall systems engineering process system (organization, people, functions, feedback/control process, models), and (5) system family research.

3. Agile requirement engineering in requirement volatility context

3.1 Definition of the problem. *R*equirements volatility is the major factor impeding success of the requirements process. Often this volatility will have major effects on such critical needs as safety. Requirements evolution is considered one of the most critical issues in developing systems; despite the recognized role of requirement engineering; requirement evolution still remains a phenomenon little understood from either qualitative or quantitative perspectives. The usual approach of trying to gather all requirements before starting design is almost associated with errors and takes too long. It has been observed in software industry that prototyping through use of agile methods is even more rapid and produces smaller amounts of code that traditional prototype (Cervi, B, 2007). Requirements changes often result from two major causes (Larsen and Buede, 2002). The first concerns revising and updating existing requirements in order to enable systems to adapt to new environments. The second is when new technology is being developed and new requirements are implemented consequently for reasons of cost or feasibility. Often, systems engineering efforts suffer from inappropriate processes which do not necessarily lead to better and mature technologies. This is often due to technology development being part of product development programs which tight schedules are basically incompatible with the time needed for creative generation and innovative solutions to technology development needs.

*3.2 Research approach***.** Two interrelated aspects of the requirements volatility problem should be considered. The first relates to the necessity of an agile requirement specific process in a volatility context. The second relates to the need for managing requirements change while simultaneously considering the impacts on such important concerns as safety.

i. Requirement process methodology. A requirement is highly dependant on the application domain. The process will be modelled in successive step of requirements with respect to domain refinement (Jackson, 1995). The requirement change process can be modelled using non monotonous logic as default reasoning. The assessment of requirement change on safety issues involves mainly the three C's: "correctness, consistency, completeness" (Zowghi and Gervasi, 2005).
ii. Requirement evolution and impacts on such important system characteristics as safety. It should be possible to carry out case study like surveys among practitioners that allow empirical investigation and determination of the integrative effects of requirements change on such important concerns as schedule, costs, performance, and maintenance releases. There is a need to develop such associated metrics as requirements maturity index, cumulative numbers of requirements change, distribution of requirements changes.

3.3 Expected results. Ideally, results of this research would enable determination of:

i. A methodology for agility in requirements engineering in the presence of requirements change;
ii. A methodology for requirements engineering in the face of volatility with safety preservation (invariants);
iii. Suggestions on how to best integrate the methodology in existing requirements management tools; and
iv. A formal risk assessment model for requirements evolution.

4. Ontology for requirements elicitation

4*.1 Definition of the problem***.** Requirements elicitation is necessarily based on the form used for requirements expression. With the advent of the Internet and electronic commerce, future business services will often be offered by autonomous and collaborating units or software agents within or across organizational boundaries through negotiation and information exchange over a distributed data network. Related efforts to develop collaborative requirement engineering emerges as an associated need.

The semantics of diverse information sources are captured by their ontology, i.e., the terms and the relationships between these sources. In many applications, the intended meaning of a term is often implicit, and understanding this in a collaborative environment necessarily is reliant upon mutual agreements and understandings. In an open environment mutual agreement is hard to achieve. Thus it is crucial for the vocabulary, which is used to describe the domain model, to be specified and maintained in such a way that other systems can process them with minimum human intervention. That is the task undertaken by ontology research that has now attracted attention from both academia and industry (Sekiuchi, R et al., 1998).

At present, it is generally difficult to develop a machine-definable ontology (vocabulary). The semantics of a term varies from one context to another and across different stakeholders. Ideally we need an approach that reduces the problem of knowing the contents and structure of many information resources to the problem of knowing the contents of domain specific ontology which a user familiar with the domain is likely to know or understand easily.

Requirements are not completely known at the start of system development. They cannot be specified completely up front in one voluminous document. But rather will evolve during analysis phases of a project and beyond. All stakeholders are involved in requirements elicitation : users, developers and customers; all see their way matured in the way the requirements are expressed from this step till maintenance; such acquired added value by the elicitation is used to improve the system instead of maintaining the myth that the requirements are to remain static.

Requirement elicitation is one of the first critical phases in requirement engineering process. Requirement process is the initial phase in systems development. The specific nature of such process is that the word “elicitation” is new as a technical terms; its equivalent does not even exist in some natural languages as French; we use some times the close definition: capture or acquire requirements . Such phase was often neglected as the requirements were only expressed; effectively, we considered were considered to be available and needs only to be expressed. For long period till the 90’s the research community was focussing on notation, methods and languages for expressing requirements. The debate was best to use for the sake for genuine expression and also for validation and verification. The debate was transported on the formal versus semi-formal specification of requirements. Most RE researchers have been concerned by such work on taxonomy of methods and adequacy of such methods and notations for expression requirements for various types of applications: in formation, systems,

The requirement engineering (RE) elicitation is one of the most critical step in RE project. Experience over the last decennia has shown that incorrect, incomplete or misunderstood requirements are the most common causes of poor quality, cost overruns and late deliveries. The ability to use an adequate approach thought a method or systematic process is therefore one of the fundamental skills in systems development. The GAO survey is a demonstration through figures on nine projects totalling about $7 millions.

A terminology (CMU) : There are many terms that are used to describe the process of understanding requirements of a systems. We use requirements engineering as a general terms encompassing all activities related to requirements. In particular, requirements engineering comprises four specific processes

Requirement elicitation: the process through which the customers, buyers, users of a system discover, reveal, articulate and understand their requirements.

Requirements analysis : the process of reasoning about the requirements that have been elicited; it involves activities such as examining requirements for conflicts and inconsistencies, combining related

requirements, and identifying missing requirements.
Requirements specification : The process of recording the requirements in one or more forms including natural language and formal, symbolic, or graphical representations; also, the product that is the document produced by that process.
Requirements validation: the process of confirming with the customer or user of the system that the specified requirements are valid, correct and complete.

In an actual situation, these four processes cannot be strictly separated and performed sequentially; they are interleaved and performed iteratively.
The term elicitation is not universally accepted for the process described; in French, there is no similar word; the term acquisition, capture is often used; some companies use gathering, expressing, formulating. Each term has a different connotation. Acquisition suppose the requirements are already there like sensor value acquisition by I/O system of a computer system. Apart from the term used , all of these terms address implicitly the elicitation term.

4.2 Research approach. The suggested research approach involves development of a shared ontology: A shared ontology can be in the form of a document or a set of machine interpretable specifications. Among possible contemporary research projects that deal with ontology-based approaches to resolving the semantic issues, the following seem especially appealing.

i. Common domain model: although participating agents share a common domain that is the basis of their cooperation, they often have different views of the domain. In order for them to collaborate, a common domain model is required to facilitate their communications.
ii. Different levels of abstraction: a flexible enterprise often requires different levels of information abstraction. At the agent level, only high-level business process and service concepts are needed to form service level agreements, i.e., contracts. At the task scheduling level, processes and services must be viewed in term of individual tasks and task interfaces (methods and conditions). At the execution level, data representation must be explicit so that data can be transformed and fused correctly.
iii. Dynamic information integration: the underlying information systems are potentially large. New services may require only parts of the information systems to be integrated. Dynamic information integration is required as which parts to be integrated for what purposes can not be determined beforehand.
iv. Service and contents description: agent services and information system contents must be formally described. The descriptions must be accessible and meaningful to all participating agents.
v. Information heterogeneity reconciliation: as flexible enterprises operate in an open environment, participating agents often use conflicting terms. In order for them to collaborate, the heterogeneity must be reconciled.

4.3 Expected results. The suggested research should result in several needed and useful outcomes.

i. Developing a requirements domain ontology environment for effective and efficient requirements elicitation will represent a considerable advance in requirements engineering. This will necessarily involve identification of appropriate support environments needed to assist ontology designers with the tasks involved in ontology management. It is envisaged that such an environment would maintain an ontology repository that can be accessed. During the design phase to enable this, tools will be available to browse and reuse the terms from the repository. When new terms need to be added, checks should be performed to see that they do not cause inconsistency in the repository. This environment should also have a set of tools that help extract ontological information that is embedded in existing systems.

ii. Develop appropriate methods and tools to support the integration of process models and information systems from multiple organizations during requirements change.
iii. Extending XML (Extended Mark-up Language) in requirements for data sources and ontology extraction and retrieval. Most CAD tools in systems design propose generation XML document for workflow.
iv. Integrate the ontology for requirements elicitation into a general framework and context to support systems engineering in a computer supported cooperative work environment.

5. A Decision Focused Framework for Life-Cycle Based Architecting and Design in SE

5.1 Definition of the Problem. Systems engineering is a multi-disciplinary problem definition and problem solving process that is implemented by people. There are as many definitions of this process as there are systems engineers with no real agreement on an underlying theory that unifies the process. Most systems engineers will agree to the following characterization of systems engineering:

i. *Focus*: a process and systems management focus that will result in the engineering of a trustworthy product or operational system that will satisfy user and customer requirements and for which these stakeholders will pay
ii. *Scope*: entire life cycle of the system, including the definition of user and customer requirements, development of the system products and enabling products, and deploying them in an operational environment. These enabling product systems include test system, deployment system, training system, operational support (logistics, maintenance, etc.), refinement system, and retirement system
iii. *Products*: Systems Engineering Management Plan, Operational Concept for the product, hierarchy of requirements documents for each key system (starting with the system-level requirements document and following the physical decomposition of the system), architectures and hierarchy of interface control documents that define the interfaces at each level of the physical decomposition
iv. *Characteristics of SE Process*: Combination of qualitative, quantitative, and executable models to examine the behavioral (functional) and system-wide (non-functional) characteristics of alternate designs and architectures.

5.2 Research Approach. A design process is characterized by a collection of decisions. In this, we use the fundamentals of decision analysis in which a decision is characterized by alternatives (what you can do – designs), values (objectives hierarchy with a quantitative value model to describe the trade-offs of the stakeholders across the key measures of effectiveness), and facts about what is known and not known. Within this context view systems engineering as a risk mitigation strategy that includes architecture, design, and testing. We must recognize that the entire process must adhere to the following principles:

i. Coherent value structure across all decisions
ii. Top-down, decentralized (distributed, asynchronous) decision making
iii. Managed by an adaptive, feedback-control process for decision making
iv. Focused, cost-effective, risk management of both the (life cycle) design and design process

5.3 Expected Results. There are at least five major hoped for outcomes of the suggested research.

i. Integration of values across all decisions for the system's life cycle
ii. Architecture and design framework for an integrated and coordinated decision-making framework with a schedule that identifies serial and concurrent decision-making activities
iii. Structure for reviews of key products that is based on the principles of feedback-control systems as well as the coordinated decision framework and is sensitive to the uncertainties

iv. Framework for risk management that is sensitive to the integrated values across the system's life cycle and the decision framework
v. Process that can be generalized to other problem solving situations

6. Architecting and Design of the Overall Systems Engineering Process System (Organization, People, Functions, Feedback/Control Process, Models)

6.1 Definition of the Problem. Systems engineering is seeing a contemporary resurgence with the recognition in the early 1990's that the many system-level failures that have occurred often were more the result of poor practices and forgotten lessons-learned, and not necessarily imperfections in the science base for technology and innovation, important as this is. This resurgence is evident within industry by the attention being devoted to the Capability Maturity Model Integration (CMMI) efforts and associated assessments for systems engineering. This has resulted in emphasis being given to defining best practices and hiring more systems engineers. In government, there also additional attention being paid as evidence by the Navy's Collaborative Engineering Environment and the Air Force's initialization of an Institute for Systems Engineering. Finally, there are an increasing number of new academic programs at both the graduate and undergraduate levels in systems engineering and systems management.

Even with these increased emphases on systems engineering and efforts to understand system-of-systems related issues, there is still a dearth of knowledge about how to tailor the overall systems engineering process as a function of the system being developed and the relationships of the organizations and humans involved in the effort. Systems engineering is a human systems problem that must be performed by multi-disciplinary teams. To achieve a successful, new system these teams overcome problems of communication, culture and technology. Examples of the key issues associated with the design of the system of people and tools that perform system engineering (SE) are:

i. Organizational structure adopted for the collection of systems engineering practitioners
   a. Alignment with the SE activities to be performed and the product design
   b. Time phasing of SE activities
   c. Interfaces among organizational elements
ii. Feedback/control structure for maintaining quality and allocating resources
   a. Few, large versus many, small feedback/control points
   b. Control actions taken
iii. Explication and use of a coherent value structure across the SE process
iv. Amount and allocation of risk management resources (including testing)

The guiding principles that dictate the research hypotheses associated with this program are:

i. The system for engineering a system or a system of systems (SE process system) is a complex, peopled process system that would greatly benefit from the application of SE
ii. Performing SE requires a recognized, explicated value structure that is used to guide all decision making throughout the SE activities
iii. Direct measure of the value achieved by the SE system is associated with the quality of the life-cycle balanced product that is produced
iv. Indirect measures of the value achieved by the SE system can be achieved by measuring the characteristics of the outputs (e.g., requirements documents), inputs, and processes.
v. Definitive measures of effectiveness are needed for the SE process system, as well as for the resulting product system and enabling product systems that are to be engineered through deployment of this SE process system.

6.2 Research Approach. The suggested research approach involves two major areas of endeavor.

i. SE Team Design – Develop a predictive simulation model of the performance of a team and team-of-teams across the spectrum of tasks required in systems engineering so that the SE process system of people, organizations, and tools can be tailored to specific systems engineering problems. Examine different strategies for problem size assigned to a team, team size, team support and training, team decision making and performance monitoring, etc. Collect data from collaborative environments for validating the results of the simulation model.

ii. Design of Organizational Interfaces and Feedback/Control Process – Develop a model that examines alternative organizational structures and interfaces (both external to customers and users as well as internal) and examine the effectiveness of quality control based upon both the organization design and the design review process. This model will allow users to evaluate different designs as well as optimize proposed designs within a limited set of parameters.

6.3 Expected Results. There are two major anticipated outcomes of successful accomplishment of the research.

i. Integration of feedback-control systems theories applied to the external behavior of systems with theories of team performance to address issues relative to why, how, what, who, when, and where. These should help us in obtaining better knowledge relative to the following issues:
   a. Predictions about time-varying environments;
   b. How work products should be controlled in a time-varying environment;
   c. How problems should be decomposed to enable the most effective combination of team performance on problem solving and communication among teams about problem solving results, as a function of problem complexity and overall domain expertise; and
   d. How communication among teams should be controlled over time in terms for format, medium, and size.

ii. One or more simulation models of with validated performance characteristics and associated statistical predictions based upon performance metrics associated with team structure for systems engineering problem definition and solution.

7. Systems Families Research

7.1 Definition of the Problem. Large systems are often formed from a variety of component systems: custom systems that are newly engineered from the “ground up;” existing commercial-off-the-shelf (COTS) systems, which are custom tailored for a particular application; and existing or legacy systems. Such related terms as systems of systems (SOS), federations of systems (FOS), federated systems of systems (F-SOS), and coalitions of systems (COS) are often used to characterize these systems. These appellations capture important realities brought about by the fact that modern systems are not monolithic. Rather, they have five characteristics initially well summarized by Mark Maier (1998) that make one of the system family designations appropriate:

1. **Operational independence of the individual systems**. A system of systems is composed of systems that are independent and useful in their own right. If a system of systems is disassembled into the constituent systems, these constituent systems are capable of independently performing useful operations by themselves and independently of one another.

2. **Managerial independence of the systems**. The component systems not only can operate independently, but generally they do operate independently in order to achieve the technological, human, and organizational purposes of the individual unit that operates the system. These component systems are generally individually acquired, serve an independently useful purpose, and often maintain a continuing operational existence that is quite independent of the larger system of systems.
3. **Geographic distribution**. Geographic dispersion of the constituent systems in a system of systems is often quite large. Often, these constituent systems can readily exchange only information and knowledge with one another, and not substantial quantities of physical mass or energy.
4. **Emergent behavio**r. The system of systems performs functions and carries out purposes that do not reside uniquely in any of the constituent systems. These behaviors arise as a consequence of the formation of the entire system of systems and are not the behavior of any constituent system. The principal purposes supporting engineering of these individual systems and the composite system of systems are fulfilled by these emergent behaviors.
5. **Evolutionary and adaptive development**. A system of systems is never fully formed or complete. Development of these systems is evolutionary and adaptive over time, and structures, functions, and purposes are added, removed, and modified as experience of the community with the individual systems and the composite system grows and evolves.

Often, appropriate missions exist for relatively large systems of systems in which there is a very limited amount of centralized command-and-control authority. Instead, a coalition of partners has decentralized power and authority and potentially differing perspectives of situations. It is useful to term such a system a “federation of systems” and sometimes a "coalition of systems." The participation of the federation or coalition of partners is based upon collaboration and coordination to meet the needs of the federation or coalition. The notions of autonomy, heterogeneity, and geographic dispersion are not independent of one another. Increasing geographic dispersion will usually lead to greater autonomy and consequently also increase heterogeneity. The Internet is perhaps the best example of a system that began under the aegis of a single sponsor, the U. S. Department of Defense, and has grown to become a federation of systems.

Support for innovation and change of all types is a desirable characteristic of these system families (Sage, 2004). Innovation includes both technological innovation and organizational and human conceptual innovation. Accomplishing this requires continuous learning, a reasonable tolerance for errors, and experimental processes to accomplish both the needed learning and the needed change. The systems fielded in order to obtain these capabilities will not be monolithic structures in terms of either operations or acquisition. Rather, they will be systems of systems, coalitions of systems, or federations of systems that are integrated in accordance with appropriate architectural constructs in order to achieve the evolutionary, adaptive, and emergent cooperative effects that will be required to achieve human and organizational purposes and to take advantage of rapid changes in technology. They can potentially accommodate: system lifecycle change, in which the life cycle associated with use of a system family evolves over time; system purpose change, in which the focus in use of the system emerges and evolves over time; and environment change, in terms of alterations in the external context supporting differing organizational and human information and knowledge needs, as well as in the technological products that comprise constituent systems.

Many conventional systems are special-purpose-built, as a mixture of commercial-off-the-shelf systems and custom developments of hardware and software. These constituents are generally provided by multiple contractors who are used to supporting a specific customer base and working under the leadership of a single vertical program management structure. For best operation, these systems should be managed as a system of systems, network of networks, federation of systems, or coalition of systems.

A system of systems generally has achieved integration of the constituent systems across communities of contractors, and sometimes across multiple customer bases, and is generally managed by more horizontally organized program management structures, such as integrated product and process development (IPPD) teams. When the IPPD team effort is well coordinated, the team is generally well able to deal with conflict issues that arise due to business, political, and other potentially competing interests (Sage and Cuppan, 2001).

7.2 Research Approach. These system family concepts have numerous implications for systems engineering and management.

i. *Grand design approach.* Contemporary organizations often treat the engineering of systems of systems or federations of systems with systems engineering protocols that are, at best, suitable only for monolithic systems. The archetype of such ill-advised protocols is the "grand design" life cycle, which is based on the waterfall model that came into prominence around 1970. A large number of problems have been encountered with grand design efforts to engineer a system. Today, the classic waterfall approach is suggested only in those rare cases where user and system-level requirements are crystal clear and unlikely to change at all during or after engineering the system, and where funding for the grand design is essentially guaranteed. This is rarely the case for major systems, especially those that are software intensive, and would be the rarest of all cases for a system of systems or federation of systems. Changing user and organizational needs and changing technologies virtually guarantee that major systems cannot be developed using the grand design approach.

ii. *Incremental and evolutionary approaches.* Two leading alternative approaches to the grand design approach for the engineering of systems were initially termed incremental and evolutionary, although the term evolutionary is now generally used to characterize both of these. In incremental development, the system is delivered in pre-planned phases or increments, in which each delivered module is functionally useful. The overall system capability improves with the addition of successive modules. The desired system capability is planned to change from the beginning, as the result of "build *N*" being augmented and enhanced through the phased increment of "build *N*+1." This approach enables a well-functioning implementation to be delivered and fielded within a relatively short time and augmented through additional builds. It also allows time for system users to thoroughly implement and evaluate an initial system with limited functionality compared to the ultimately desired system. Generally, the notion of preplanning of future builds is strong in incremental development. As experience with the system at build *N* is gained, requirements changes for module *N*+1 may be more easily incorporated into this, and subsequent, builds.

Evolutionary lifecycle development is similar in approach to its incremental complement; however, future changes are not necessarily pre-planned. This approach recognizes that it is impossible to initially predict and set forth engineering plans for the exact nature of these changes. The system is engineered at build *N*+1 through reengineering the system that existed at build *N*. Thus, a new functional system is delivered at each build, rather than obtaining build *N*+1 from build *N* by adding a new module. The enhancements to be made to obtain a future system are not determined in advance, as in the case of incremental builds. Evolutionary development approaches can be very effective in cases where user requirements are expected to shift dramatically over time, and where emerging and innovative technologies allow for major future improvements. They are especially useful for the engineering of unprecedented systems that involve substantial risk and allow potentially enhanced risk management. Evolutionary development may help program managers adjust to changing requirements and funding priority shifts over time since new functionality introductions can be advanced or delayed in order to accommodate user requirements and funding changes. Open, flexible, and adaptable system architecture is central to the notion of evolutionary and emergent development. These are major elements in the contemporary U.S.. Department of Defense Initiatives in evolutionary acquisition and such issues as Network Centric Warfare.

7.3 Expected Results. Research in this area would seek to develop and apply notions from such areas as complex adaptive system and knowledge management and would seek to develop more of a methodological basis for system family architecting and design. Development of a methodological basis for the design and architecting of system families would do much to enhance present abilities to design loosely coupled and virtual organizations and to enable better architectures for these enterprises that would do much to support interoperability and integration.

CONCLUSIONS

Substantial research is required to enable systems engineering to continue to progress and to provide the necessary productivity improvements expected by those who purchase systems engineering products and services. This paper describes five topics that the authors believe are of significant importance to the future of systems engineering. Hopefully these ideas can generate more discussion and activity among those involved in the research and practice of systems engineering.

References

Cervi, B.: Race for flexibility , Briefing Agile Production *Journal Manufacturing Engineer* Volume 86, Issue 4, Aug.-Sept. 2007 Page(s):14 – 15

Friedman, G. and Sage, A. P. (2004), "Case Studies of Systems Engineering and Management in Systems Acquisition," *Systems Engineering*, Vol. 7, No. 1, pp.84-97

Gotel, O., Finkelstein, A. An Analysis of the Requirements Traceability Problem Proc. of First *International Conference on Requirements Engineering*, 1994, pages 94-101

Honour, E. and Buede, D. (1998). SE Research Agenda. Private communication

Jackson, M. (1995), *Software Requirements and Specification: a Lexicon of Practice, Principles and Prejudices,* Addison-Wesley.

Larsen, R. and Buede, D. (2002), "Theoretical Framework for the Continuous Early Validation (CEaVa) Method," *Systems Engineering*, Vol. 5, No. 3, pp. 223-241.

Maier, M. W. (1998), “Architecting Principles for Systems-of-Systems,” *Systems Engineering*, Vol. 1, No. 4, pp. 267-284.

Messaadia, M, Sahraoui, AEK (2005) “PLM as linkage process in a systems engineering framework” *International Journal of Product Development*, Vol.4, N°3/4, pp.382-395, 2007

Sage, A. P. (2004), " Sustainable Development: Issues in Information, Knowledge, and Systems Management ",. Information-Knowledge-Systems Management, Volume 1 , Issue 3,4 (December 1999) Pages: 185 - 223

Sage, A. P. and Cuppan, C. D. (2001), "On the Systems Engineering and Management of Systems of Systems and Federations of Systems," *Information, Knowledge, and Systems Managemen*t, Vol. 2, No. 4, pp. 325-345.

Sahraoui, A.E.K (1999) "System engineering: a discipline for unifying engineering education", *INCOSE INSIGHT*, Issue 1, Vol.2, pp.19-21, Spring 1999

Sekiuchi, R., Aoki, C., Kurematsu, M., and Yamaguchi, T., (1998): DODDLE : A Domain Ontology Rapid Development Environment., In *Proc of PRICAI 98* (Pacific Rim International Conferences on Artificial Intelligence*)*.

Zowghi, D. and Gervasi, V. (2005) "On the Interplay Between Consistency, Completeness, and Correctness in Requirement Evolution," ACM Transactions on Software Engineering and Methodology, Volume 14 , Issue 3 (July 2005) Pages: 277 - 330