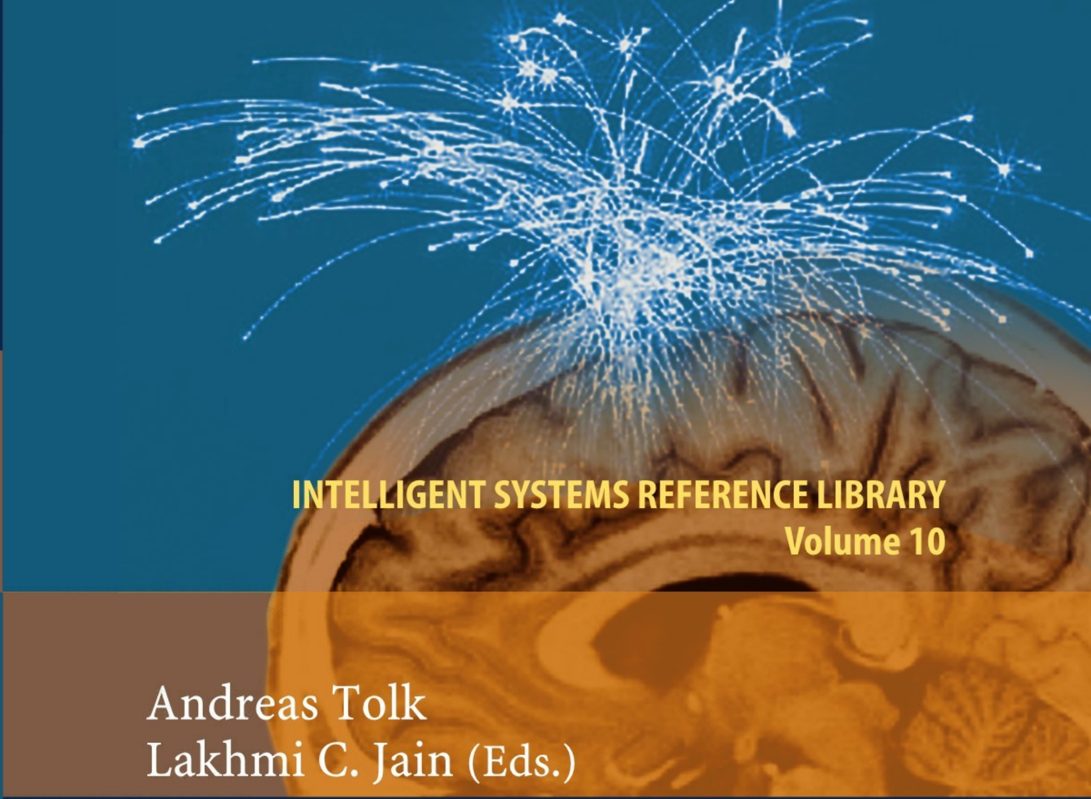

INTELLIGENT SYSTEMS REFERENCE LIBRARY
Volume 10

Andreas Tolk
Lakhmi C. Jain (Eds.)

Intelligence-Based Systems Engineering

Springer

Andreas Tolk and Lakhmi C. Jain

Intelligence-Based Systems Engineering

Intelligent Systems Reference Library, Volume 10

Editors-in-Chief

Further volumes of this series can be found on our homepage: springer.com

Vol. 1. Christine L. Mumford and Lakhmi C. Jain (Eds.)
Computational Intelligence: Collaboration, Fusion and Emergence, 2009
ISBN 978-3-642-01798-8

Vol. 2. Yuehui Chen and Ajith Abraham
Tree-Structure Based Hybrid Computational Intelligence, 2009
ISBN 978-3-642-04738-1

Vol. 3. Anthony Finn and Steve Scheding
Developments and Challenges for Autonomous Unmanned Vehicles, 2010
ISBN 978-3-642-10703-0

Vol. 4. Lakhmi C. Jain and Chee Peng Lim (Eds.)
Handbook on Decision Making: Techniques and Applications, 2010
ISBN 978-3-642-13638-2

Vol. 5. George A. Anastassiou
Intelligent Mathematics: Computational Analysis, 2010
ISBN 978-3-642-17097-3

Vol. 6. Ludmila Dymowa
Soft Computing in Economics and Finance, 2011
ISBN 978-3-642-17718-7

Vol. 7. Gerasimos G. Rigatos
Modelling and Control for Intelligent Industrial Systems, 2011
ISBN 978-3-642-17874-0

Vol. 8. Edward H.Y. Lim, James N.K. Liu, and Raymond S.T. Lee
Knowledge Seeker – Ontology Modelling for Information Search and Management, 2011
ISBN 978-3-642-17915-0

Vol. 9. Menahem Friedman and Abraham Kandel
Calculus Light, 2011
ISBN 978-3-642-17847-4

Vol. 10. Andreas Tolk and Lakhmi C. Jain
Intelligence-Based Systems Engineering, 2011
ISBN 978-3-642-17930-3

Andreas Tolk and Lakhmi C. Jain

Intelligence-Based Systems Engineering

Springer

Prof. Andreas Tolk
Engineering Management & Systems Engineering
242B Kaufman Hall
Old Dominion University
Norfolk, VA 23529
USA
E-mail: atolk@odu.edu

Prof. Lakhmi C. Jain
School of Electrical and Information Engineering
University of South Australia
Adelaide
Mawson Lakes Campus
South Australia SA 5095
Australia
E-mail: Lakhmi.jain@unisa.edu.au

ISBN 978-3-642-17930-3 e-ISBN 978-3-642-17931-0

DOI 10.1007/978-3-642-17931-0

Intelligent Systems Reference Library ISSN 1868-4394

Typeset & *Cover Design:* Scientific Publishing Services Pvt. Ltd., Chennai, India.

Printed on acid-free paper

9 8 7 6 5 4 3 2 1

springer.com

Preface

The International Council on Systems Engineering (INCOSE) defines Systems Engineering as an interdisciplinary approach and means to develop successful systems. It focuses on defining the customers needs and requirements early in the development cycle. It then documents the requirements. It then proceeds with the design synthesis and system validation and develops an overview of the complete problem which involves Manufacturing, Operations, Cost & Scheduling. The Performance, Training & Support, Testing, and Disposal are then developed. Systems Engineering integrates all of the disciplines and specialty groups into a joint team effort to form a structured development process which proceeds from the concept stage of production to full final operation. The full Systems Engineering operation considers both the business and the technical needs of all customers. The goal is to provide a quality product that meets the user needs and hopefully without unwanted surprises in the completed item.

In the present time, these activities and processes are increasingly supported by means of Information Technology (IT). Support using IT always leads to the question of how much such processes can be either automated or semi-automated. In other words: is it possible to increase the quality of systems by using intelligence-based systems engineering. The intention of this book is to answer the questions such as what emerging methods and solutions are able to use intelligence-based systems engineering, what current solutions already exist, what theoretic constraints are known, and other questions ranging between theory and practice. The chapters contain contributions from conferences, research, PhD theses, and the experience of the experts in this area. In this book, we establish a research agenda and begin to fill the gaps in this body of knowledge.

We hope to gain the support of practitioners and scholars by this volume. It is also hoped to help researchers identify domains of interest and to develop systems engineering to an even higher level.

Andreas Tolk
USA

Lakhmi C. Jain
Australia

Contents

List of Contributors

Kevin MacG. Adams
National Centers of System of
Systems Engineering
Old Dominion University
Norfolk, VA, USA

Khaldoon Al-Zoubi
Dept. of Systems and Computer
Engineering Carleton University
Ottawa, ON. K1S 5B6
Canada

Ian Bailey
Model Futures Limited
London, United Kingdom

Zuhair Bandar
School of Computing, Mathematics &
Digital Technology
Manchester Metropolitan University
Manchester M1 5GD
United Kingdom

Yolanda Blanco-Fernández
Department of Telematics
Engineering
University of Vigo
Vigo, Spain

Keeley Crockett
School of Computing, Mathematics &
Digital Technology
Manchester Metropolitan University
Manchester M1 5GD
United Kingdom

Saikou Y. Diallo
Virginia Modeling Analysis and
Simulation Center
Old Dominion University
Norfolk, VA, USA

Michele Fumarola
Delft University of Technology
The Netherlands

C. Anthony Hunt
Department of Bioengineering and
Therapeutic Sciences
University of California
San Francisco, CA, USA

Dietmar Jannach
Technical University of Dortmund
Baroperstraße 301
D-44227 Dortmund, Germany

Raja Jurdak
CSIRO ICT Centre
and University of
Queensland/School of ITEE
Brisbane, QLD
Australia

Charles B. Keating
Engineering Management and
Systems Engineering
Old Dominion University
Norfolk, VA, USA

Martín López-Nores
Department of Telematics
Engineering
University of Vigo
Vigo, Spain

Saurabh Mittal
Dunip Technologies
Tempe, AZ, USA

Miklos Nagy
KMI, Open University
Milton Keynes, MK7 6AA
England, UK

Oliver Obst
CSIRO ICT Centre
Sydney, NSW
Australia

James O'Shea
School of Computing, Mathematics &
Digital Technology
Manchester Metropolitan University
Manchester M1 5GD
United Kingdom

José J. Padilla
Virginia Modeling Analysis and
Simulation Center
Old Dominion University
Norfolk, VA, USA

José J. Pazos-Arias
Department of Telematics
Engineering
University of Vigo
Vigo, Spain

Mamadou D. Seck
Delft University of Technology
The Netherlands

Andres Sousa-Poza
Engineering Management and
Systems Engineering
Old Dominion University
Norfolk, VA, USA

John F. Sowa
VivoMind Research, LLC
Croton on Hudson, New York
USA

Claudia Szabo
Department of Computer Science
National University of Singapore
Singapore

Andreas Tolk
Engineering Management and
Systems Engineering
Old Dominion University
Norfolk, VA, USA

Philip Valencia
CSIRO ICT Centre
Brisbane, QLD
Australia

Maria Vargas-Vera
Computing Department
Open University
Milton Keynes, MK7 6AA
England, UK

Alexander Verbraeck
Delft University of Technology
The Netherlands

Gabriel A. Wainer
Dept. of Systems and Computer
Engineering Carleton University
Ottawa, ON. K1S 5B6
Canada

X. Rosalind Wang
CSIRO ICT Centre
Marsfield, NSW
Australia

Levent Yilmaz
Department of Computer Science and
Software Engineering
Auburn University
Auburn, AL, USA

Resumes of Contributing Authors

Kevin MacG. Adams is a Principal Research Scientist at the National Centers for System of Systems Engineering (NCSoSE) of Old Dominion University in Norfolk, Virginia. He holds a Ph.D. in systems engineering from Old Dominion University, dual Master's degrees in Materials Engineering and Naval Architecture and Marine Engineering from the Massachusetts Institute of Technology, and a Bachelor's degree in Ceramic Engineering from Rutgers University. His research focuses on system of systems engineering, systems engineering methodologies, software engineering project management frameworks, the philosophy of science, and the use of enterprise architectures. He is a retired Navy submarine engineering duty officer, a senior member of the Institute of Electrical and Electronics Engineers (IEEE), a member of the American Association of University Professors, and the United States Naval Institute.

Khaldoon Al-Zoubi is a Ph.D. student in Electrical Engineering within the Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada. He is also a senior software analyst and programmer with over 13 years of industry experience occupying a number of seniority and leadership positions. His industry experience spreads over wide range of areas such as embedded software and mobility, air-traffic software management and telecommunications, and security software for explosive and narcotics detections.

Ian Bailey founded Model Futures in 2004 to provide software, consultancy and training in information management. He specializes in enterprise architecture and ontology. He was the technical lead in the development of the UK MOD Architecture Framework (MODAF) and co-developed the SOA views for the NATO Architecture Framework. He is the lead modeler in the multi-nation IDEAS ontology project, which targets to become a common foundation ontology for defense enterprise architecture. Previous to working with MODAF, Ian was editor of the ISO10303-233 systems engineering standard. Most of his professional life prior to Model Futures was spent integrating and re-engineering large scale information systems for customers such as Amec, BAE Systems, BP, Shell and Volvo. He has a Ph.D. in data mapping and a first degree in mechanical engineering. He is a fellow of the Institute of Engineering and Technology (IET).

Zuhair Bandar is a Reader in Intelligent Systems at MMU. He received his Ph.D. in AI and Neural Networks from Brunel University, his M.Sc. in Electronics from the University of Kent and his B.Sc. in Electrical Engineering from Mosul University. He is a co-founder of the ISG and his research interests include the application of AI to psychological profiling. He is the Technical Director of Convagent Ltd, an MMU spinout company which provides business rule automation with natural language interfaces using conversational agents.

Yolanda Blanco-Fernández was born in Orense, Spain in 1980. She received the Telecommunications Engineering Degree from the University of Vigo in 2003, and the Ph.D. degree in Computer Science from the same University in 2007. Nowadays, she is an assistant professor in the Department of Telematics Engineering, teaching in courses related to network management systems, multimedia services and operating systems. Her main research activity involves development of personalization services for Interactive Digital TV and e-commerce, by applying technological foundations borrowed form the Semantic Web, Web 2.0 and cloud computing.

Keeley Crockett is a Senior Lecturer at MMU. She received her Ph.D. in Machine Learning from MMU and her B.Sc. in Computation from the University of Manchester Institute of Science and Technology. She is a committee member of the IEEE Women into Computational Intelligence Society and a full member of the IEEE Computational Intelligence Society. Her main research interests include fuzzy decision trees, applications of fuzzy theory, and data mining. She is a knowledge engineer and founding member of Convagent Ltd.

Saikou Y. Diallo is a Research Assistant Professor with the Virginia Modeling, Analysis and Simulation Center (VMASC) at Old Dominion University, Suffolk, VA. He received his B.Sc. in Computer Engineering and his M.S. and Ph.D. in Modeling and Simulation from Old Dominion University. He is author of several awarded articles on interoperability and composability. His research focus is on the theory of interoperability.

Michelle Fumarola is a Ph.D. student at the Systems Engineering Group of Delft University of Technology. His Ph.D. research is focused on developing simulation games for decision making with a strong visual component.

C. Anthony Hunt is Professor of Bioengineering and Therapeutic Sciences at the University of California in San Francisco, where he directs the BioSystems Group. He earned a Ph.D. in pharmaceutical chemistry from the University of Florida and B.Sc. degrees in both chemistry and applied biology from the Georgia Institute of Technology. He chairs the BioSystems Group that develops and uses advanced modeling and simulation methods to achieve deeper insight into the networked micromechanisms that link molecular level events with higher level phenomena and operating principles at cell, tissue, organ, and organism levels in the presence and absence of interventions. He is a member of the Editorial Boards of Simulation, Transactions of the SCS, the International Journal of Knowledge Discovery in Bioinformatics, and the Journal of Computational Biology and Bioinformatics Research. He is a Fellow of the American Association for the Advancement of Science and the American Association of Pharmaceutical Scientists, a Director of The McLeod Modeling and Simulation Network, and a member of several professional scientific and engineering associations.

Dietmar Jannach is a full professor at Technische Universität Dortmund, Germany and the head of the e-Services Research Group. His research interests include interactive recommender systems and conversational preference elicitation, engineering of knowledge-based systems and web applications as well as the application of Artificial Intelligence in industry. He has authored and co-authored more than 100 scientific papers in these areas and published papers in journals such as Artificial Intelligence, AI Magazine, IEEE Intelligent Systems and on conferences such as IJCAI and ECAI.

Raja Jurdak is a Principal Research Scientist at CSIRO ICT Centre. He holds a Ph.D. in Information and Computer Sciences and an MS in Computer Engineering from the university of California, Irvine and a BE in Computer and Communications Engineering from the American University of Beirut. He is an adjunct Associate Professor at University of Queensland's School of Information Technology and Electrical Engineering. He is also a member of the IEEE and the IEEE Communications Society. His current research interests focuses on modeling, optimization, and real world deployments of energy-efficient and highly responsive sensor networks. He has over 40 peer-reviewed journal and conference publications, as well as a book published by Springer titled Wireless Ad Hoc and Sensor Networks: A Cross-Layer Design Perspective.

Charles B. Keating is a Professor of Engineering Management and Systems Engineering at Old Dominion University located in Norfolk, Virginia. He also serves as the Director for the National Centers for System of Systems Engineering (NCSoSE) where his research is focused on development and testing of theory, methodologies, and technologies to more effectively deal with complex system problems. Prior to joining the university, Dr. Keating had over 12 years of experience in command and technical engineering management positions in the U.S. Army, Texas Instruments, and Newport News Shipbuilding. Dr. Keating holds a B.Sc. in Engineering from the United States Military Academy (West Point), an M.A. in Management from Central Michigan University, and a Ph.D. in Engineering Management from Old Dominion University. His current research interests include: System of Systems Engineering, Complex System Problem Domains, and Complex System Governance.

Martín López-Nores was born in Pontevedra, Spain in 1980. He received the Telecommunications Engineering Degree from the University of Vigo in 2003, and the Ph.D. degree in Computer Science from the same University in 2006. Nowadays, he is an associate professor in the Department of Telematics Engineering, teaching in courses related to computer networks, operating systems and information services. Starting from works on applied formal specification techniques, his research interest have evolved to embrace the design and development of interactive services for a range of consumer electronics devices, the design and evaluation of communication protocols and innovative applications for mobile ad-hoc networks, and the management of health-related data in semantics-based recommender systems and pervasive computing environments.

Saurabh Mittal is the President and founder of Dunip Technologies, Tempe, AZ, USA and is also a Research Scientist at US Air Force Research Lab (AFRL), 711th Human Performance Wing, for L-3 Communications, Link Simulation and Training Branch in Mesa, AZ. His work at AFRL involves multiformalisms, cognitive modeling and net-centric systems engineering. He holds a Ph.D. (2007) and an M.S. (2003), both in Electrical and Computer Engineering from the University of Arizona, Tucson. He was recognized with Golden Eagle Award, highest recognition to any civilian by Joint Interoperability Test Command and US Air Force, supporting the US Warfighter for his work on Generic Network System Capable of Planned Expansion (GENETSCOPE) in 2006. His research focuses on Discrete Event modeling using DEVS Formalism, net-centric system of systems engineering with DEVS Unified Process, executable architectures using Department of Defense Architecture Framework (DoDAF), simulation-based computing and large scale M&S infrastructures using Service oriented architecture. He is a member of Institute of Electrical and Electronics Engineers (IEEE), Association of Computer Machinery (ACM), and Society for Modeling and Simulation International (SCS).

Miklos Nagy is a Ph.D. candidate at the Open University's Knowledge Media Institute. His research interests are Uncertain Reasoning, Ontology Mapping, Multi-agent systems and information integration using Semantic Web technologies. His current research focuses on the development of intelligent multi-agent systems that can exploit the emerging Semantic Web's large-scale data. Miklos Nagy received his MSc in Information Engineering from the University of Miskolc, Hungary.

Oliver Obst is Research Scientist at the Adaptive Systems Team at the Australian Commonwealth Scientific and Research Organization (CSIRO) in Sydney. He holds a Ph.D. and a M.Sc. in computer science, both from the University of Koblenz-Landau in Koblenz, Germany. He is affiliated as an honorary associate with the School of Information Technologies of the University of Sydney. His research fields are information processing in neural networks, the application of information-theory to guide self-organization in complex systems, the representation of sensory information, as well as the emergence of coding for both technical and biological systems. His work involves development of new machine learning algorithms and architectures, as well as their applications to real world problems, for example in fault detection in distributed systems, such as smart electrical grids or sensor networks. He is a member of the International Neural Network Society (INNS), and the German Informatics Association (GI e.V.).

James O'Shea is a senior lecturer at Manchester Metropolitan University (MMU). He received his B.Sc. in Chemistry from Imperial College. He worked in computer R&D at International Computers and as an independent consultant under the UK Microelectronics Applications Project until 1985. After joining MMU, he developed a research interest in AI and co-founded the Intelligent Systems Group (ISG). In addition to his work in the field of CAs, he is one of the inventors of the Silent Talker lie detector, which has attracted worldwide interest.

Jose J. Padilla is a Post Doctoral Research Associate with the Virginia Modeling, Analysis and Simulation Center (VMASC) at Old Dominion University, Suffolk, VA.

He received his Ph.D. in Engineering Management from Old Dominion University. He also holds a B.Sc. in Industrial Engineering from the Universidad Nacional de Colombia, Medellín, and an M.B.A. International Business from Lynn University, Boca Raton, Florida. His research interests are on the nature of the process of understanding and how it contributes to the perception of complexity for a human or a computer agent.

José J. Pazos-Arias is Full Professor at Department of Telematics Engineering at the University of Vigo (Spain). He received his degree in Telecommunications Engineering from the Technical University of Madrid (Spain-UPM) in 1987, and his Ph.D. degree in Computer Science from the Department of Telematic Systems Engineering at the same University in 1995. He is the director of the Interactive Digital TV Laboratory, which is currently involved with national and international projects, receiving funds from both public institutions and industry. With the aim of combining the power of semantic reasoning technologies and the participation phenomena arising in the knowledge society, he is currently involved in the use of social-semantic technologies to assist the users when it comes to facing complex decision takings in the cloud. In this regard, he is highly interested in gaining deeper knowledge in social network analysis and emergent semantics.

Mamadou D. Seck received his Ph.D. degree from the Paul Cezanne University of Marseille and his MS and M.Eng Degrees from Polytech' Marseille, France. He is currently an Assistant Professor in the Systems Engineering section at the Technology, Policy, and Management department of Delft University of Technology. His research interests include modeling and simulation formalisms, dynamic data driven simulation, human behavior representation and social simulation, and agent directed simulation.

Andres Sousa-Poza is an Associate Professor of Engineering Management and Systems Engineering at Old Dominion University. He holds a B.Sc. in Mechanical Engineering from the University of Cape Town, South Africa, and M.S. and Ph.D. degrees in Engineering Management from the University of Missouri-Rolla, USA. He is the co-founder and present chair of the Management and Engineering in Complex Situations Forum (MECS Forum). He is affiliated with the National Centers for System of Systems Engineering (NCSoSE) as a senior researcher.

John F. Sowa spent thirty years working on research and development projects at IBM and is a cofounder of VivoMind Research, LLC. He has a B.Sc. in mathematics from MIT, an MA in applied mathematics from Harvard, and a Ph.D. in computer science from the Vrije Universiteit, Brussel. He is a fellow of the American Association for Artificial Intelligence. With his colleagues at VivoMind, he has been developing novel methods for using logic and ontology in systems for reasoning and language understanding. He designed the language of conceptual graphs, which has been adopted as one of the three normative dialects of the ISO/IEC standard for Common Logic.

Claudia Szabo is a Research Assistant at the Department of Computer Science, National University of Singapore. She received her B.Sc. from the "Politehnica" University of Bucharest and is a Ph.D. candidate in Modeling and Simulation at the

National University of Singapore. Her research evaluates simulation composability and interoperability, with a focus on formal validation. A unique aspect of her work is to enable trade-off analysis between validation accuracy and computational cost. She is the author of several awarded articles of composability and validation, including the 2009 ACM SIGSIM Best Ph.D. Student Paper Award.

Andreas Tolk is Associate Professor for Engineering Management and Systems Engineering at Old Dominion University in Norfolk, Virginia. He holds a Ph.D. and an M.S. in computer science, both from the University of the Federal Armed Forces of Germany in Munich. He is affiliated as a Senior Research Scientist with the National Centers for System of Systems Engineering (NCSoSE) in Norfolk, Virginia, and the Virginia Modeling Analysis and Simulation Center (VMASC) in Suffolk, Virginia. His research focuses on integratability and composability of model-based solutions and modeling and simulation based systems engineering. He is member of the American Society for Engineering Management (ASEM), Association for Computing Machinery (ACM), Institute of Electrical and Electronics Engineers (IEEE), Military Operational Research Society (MORS), National Defense Industrial Association (NDIA), Simulation Interoperability Standards Organization (SISO), and Society for Modeling and Simulation International (SCS). He was recognized with the Excellence in Research Award of the Frank Batten College of Engineering and Technology and received the first Technical Merit Award of SISO.

Philip Valencia is a Research Engineer at the Autonomous Systems Laboratory within the Commonwealth Scientific and Research Organization (CSIRO) ICT Centre. He holds bachelor degrees in Engineering (Electronic) and IT from the Queensland University of Technology, Brisbane, Australia and is undertaking Ph.D. studies at the University of Queensland. He has eight years research experience with wireless sensor network technologies as well as background in machine learning which he has brought together to research distributed online learning for wireless sensor and actuation networks.

Maria Vargas-Vera is a Lecturer in Computing at the Open University, England UK. She received her Ph.D. from the Artificial Intelligence Department at Edinburgh University and her MSc in Computer Science from the National University of Mexico (UNAM). She was awarded Fellow of the British Computer Society (FBCS) from November 2009. Her current research focuses on Automatic Construction of Ontologies from Text, Ontology Mapping and E-Learning Applications using Semantic Web Technologies. Maria Vargas-Vera has published many research papers in prestigious journals and international conferences and she is a member of program committees of international conferences and workshops. Also, she is an Associated Editor of the International Journal of Knowledge and Learning (IJKL) and the Journal of Emerging Technologies in Web Intelligence (JETWI).

Alexander Verbraeck got his M.Sc. in mathematics in 1987 and his Ph.D. in 1991 from Delft University of Technology in the Netherlands. Until 1992 he also had his own software company, focusing on consultancy and software development for educational institutes. He worked as assistant professor in information systems until 1995, when he was appointed associate professor in the systems engineering group of the faculty of Technology, Policy and Management (TPM) of TU Delft. He is the chair of the Systems Engineering research group and he has been department chair of the Department of Information, Communication and Systems of the TPM Faculty of TU Delft. He was the original author and program manager of the BETADE strategic research program of TU Delft. He has also been appointed part-time research professor at the R.H. Smith School of Business of the University of Maryland, USA.

Gabriel Wainer is Associate Professor at the Department of Systems and Computer Engineering of Carleton University, Ottawa, ON, Canada. He is head of the Advanced Real-Time Simulation lab, located at Carleton University's Centre for advanced Simulation and Visualization. He received the M.Sc. (1993) and Ph.D. (1998) in Computer Science from the Universidad de Buenos Aires, Argentina, and IUSPIM (now Polytech de Marseille), Université Paul Cézanne, Aix-Marseille III, France. Previously, he was Assistant Professor in the Computer Sciences Department of the Universidad de Buenos Aires, and held visiting positions in numerous places, including the Arizona Center of Integrated Modeling and Simulation (ACIMS, University of Arizona), Laboratory of Systems Sciences of Marseille (LSIS-CNRS), University of Nice, Polytech de Marseille, INRIA Sophia-Antipolis (France). He is the Vice-President Publications, and was a member of the Board of Directors of the Society for Computer Simulation International (SCS). He is the author of three books and over 200 articles in different venues. He has collaborated in the organization of over 100 conferences and is co-founder of the SIMUTools Conferences. He has been the recipient of various awards, including the IBM Eclipse Innovation Award, a Leadership award by SCS, and various Best Paper awards. He has been awarded Carleton University's Research Achievement Award and the First Bernard P. Zeigler DEVS Modeling and Simulation Award.

Rosalind Wang is currently a research scientist in the ICT Centre, CSIRO. She received her Ph.D. in Mechatronics Engineering from the University of Sydney, Australia in 2009. Her research interests are machine learning, graphical models, and pattern recognition.

Levent Yilmaz is Associate Professor of Computer Science and Software Engineering and holds a joint appointment with the Industrial and Systems Engineering department at Auburn University. He received his B.Sc. degree in Computer Engineering and Information Sciences from Bilkent University and M.S. and Ph.D.. degrees in Computer Science from Virginia Tech. His research interests are in Modeling & Simulation, Agent-directed Simulation, and Complex Adaptive Systems, focusing in theory and methodology of modeling and simulation to advance scientific discovery and theory formation, and to develop robust decision support systems and models of socio-technical, cognitive, and cultural systems, such as science of science and innovation policy. He is a member of the Board of Directors of SCS and is the Editor-in-Chief of Simulation: Transactions of the Society for Modeling and Simulation International. He is member of ACM, IEEE Computer Society, Society for Computer Simulation International (SCS), and Upsilon Pi Epsilon.

Chapter 1
Towards Intelligence-Based Systems Engineering and System of Systems Engineering

Andreas Tolk, Kevin MacG. Adams, and Charles B. Keating

Department of Engineering Management and Systems Engineering
National Centers of System of Systems Engineering
Old Dominion University
Norfolk, Virginia 23508-2563, USA
atolk@odu.edu, kmadams@odu.edu, ckeating@odu.edu

Abstract. This introductory chapter defines intelligence-based systems with focus on semantic systems, simulation systems, and intelligent agents. Semantic systems define the foundation to communicate systems engineering challenges using logic, simulation systems introduce the dynamic component, and intelligent agents can introduce alternatives roles. It then gives an overview of traditional systems engineering as well as system of systems engineering showing the need to emphasize the system of systems perspective in modern engineering approaches. Finally, both views are aligned, providing a scope for intelligence-based systems engineering and the contributions of the following book chapters are summarized in relationship to this scope.

Keywords: intelligent agents, ontology, semantic system, simulation system, system of systems engineering, systems engineering.

1 Introduction

The definition of *insanity* as "doing the same thing over and over again and expecting different results" is attributed to Albert Einstein. In contrast, a collective definition for *intelligence* is the ability to comprehend, to understand and profit from experience, or to make sense out of the environment and react appropriately. In the light of these two extremes, this introductory chapter defines what intelligence-based systems are, and what this means for systems engineering and systems of systems engineering.

Starting with a summary of the state of the art, as among others identified by Buede [1], it can be observed that most of our current systems have been designed starting with a set of well defined requirements. These requirements are often based on operational concepts that identify context and external systems and that are used to derive (a) input and output requirements that identify what a system shall accept and produce, (b) system-wide and technology requirements that are building a set of operational constraints, (c) trade-off requirements that allow optimizing system design decisions within these constraints, and (d) qualification requirements that allow validation and verification to be conducted. These requirements lead to building a functional architecture describing the capabilities of the system, a physical architecture that describes the resources that comprise the system, and finally an allocated

A. Tolk and L.C. Jain (Eds.): Intelligence-Based Systems Engineering, ISRL 10, pp. 1–22.
springerlink.com

architecture that merges the functional and the physical view, including interface design, integration and qualification. The result is a well-defined system that has a well defined behavior for all identified input constellation in the form of expected output produced. As a rule, the capabilities defined in the functional architecture are fixed. The system will do the same thing over and over again. Under many circumstances, this is exactly what we would want. Nobody wants to push down the brake pedal of a car expecting anything else but that the car stops. We expect the same results. However, what if the environment changes? What if the world in which a system was originally defined no longer exists?, like we currently see it in so many military systems that were defined at the time of the Cold War, but still have to be used today? Simply expecting the system to change its behavior qualifies as insanity, so we need intelligent systems that are able to comprehend, understand and profit from experience.

The next section will define intelligence-based systems. Following these definitions and examples, the third section will evaluate the relation of such systems with systems engineering. The fourth section will do the same for the new and emerging field of system of systems engineering that adds at least one additional layer of complexity to the challenges to be addressed. Finally, the last section will describe the contributions comprised in this book in the light of these findings.

2 Intelligence-Based Systems

Intelligence-based systems should not be confused with the often narrowly used term intelligence system, which refer to a variety of Artificial Intelligence (AI) methods, such as neural networks, evolutionary algorithms, expert systems, diagnostic systems, symbolic AI, and other related topical areas. These systems are limited to AI applications, and intelligent systems engineering describes the engineering of such intelligent systems, not the use of intelligence to support systems engineering. The scope we take in this chapter – and in this book in general – includes the design and engineering of such intelligent systems, but is not limited to this view. We are interested in merging the state of the art of intelligence as it can be provided via AI methods to support systems engineering and system of systems engineering. How can these three aspects be of mutual support, resulting in better systems that are able to comprehend, understand and profit from experience. This is the objective of intelligence-based systems engineering: to base systems and their design on AI methods to build better systems.

2.1 Characteristics of Intelligence-Based Systems

In order to support this objective of intelligence-based systems engineering, it is first important to better understand the characteristic properties of intelligence-based systems. The following list is neither complete nor exclusive, but it reflects the collective definition of various views on AI, intelligence-based solutions, model-based prediction and control, and similar contributions. Figure 1 depicts these characteristics that are used in the collective definition, which are self-explaining, robust, fault tolerant, adaptive, self-optimizing, deductive, learning, cooperative, autonomous, and agile. As we will see, these terms have partly overlapping definitions and have to be understood in the context of the collective definition, which means that not all definitions use all terms.

Fig. 1. Characteristic Properties of Intelligence-based Systems

Self-explaining doesn't mean that the system is obvious without any explanation necessary, but that the system can explain how it came to a certain decision. In traditional systems, the system behavior does not change. If a system is able to modify its behavior, it is often needed to understand how and why a decision has been made by the system. The explanation component of expert systems used for diagnosis, which traditionally could be generated by tracing the line of reasoning used by the underlying inference engine to answer the questions: "Why is your answer to the question the one you recommend?" For systems that are able to modify themselves being able to explain their reason is mandatory to ensure credibility.

Robust as a characteristic property of a system means that the system behaves well and adequate not only under ordinary conditions, but also under unusual conditions that stress the original requirements and derived assumptions. In other words, robust systems do not break easily, but are able to continue to behave well even under variant circumstances that could lead to failure of system.

Fault tolerant systems behave well and continue to adequately perform even if one or more of its internal system components fail or break. It may be important to differentiate between a fault, which is a defect in the system that can cause an error, which is a subset of the system status that may lead to system failure, which is a deviation in actual system behavior and its desired behavior according to the requirements.

Adaptive systems in general react to changes, in particular to changes in the environment or the context of the system. Whenever the environment or context of the system changes the system itself changes as well in order to accommodate these changes. As a consequence, adaptive systems behave well and adequate even in changing environments.

Self-organizing systems organize their internal components and capabilities in new structures without a central or an external authority in place. These new structures can be temporal and spatial. In some cases, instead of self-organizing the term self-optimizing is used synonymously, although not all self-organizing structures represent the optimal structure, but the assumption is that self-organizing systems are organizing themselves to become better.

Deductive systems are well known from mathematics: based on a set of axioms and rules, they can deduct new insights by applying the rules to the axioms as well as to the resulting new facts. This is done using an underlying inference engine. Applying these ideas, deductive systems can discover new facts that they can use for their decision process on how to modify themselves to behave well and adequate.

Learning systems generally observe the achieved results and compare them with the desired outcome. Using methods such as reinforcement learning, decisions that led to positive results are enforced while those with negative results are avoided. Learning can also occur by observing other systems and the results of their activities. In every case, learning is connected with the observation of cause and effects.

Cooperative systems expose social capabilities. This means that cooperative systems interact with other systems – and potentially humans as well – via some kind of communication language. This interaction is not limited to pure observation, but such a system can exchange plans, distribute tasks, etc. Whiteboard technologies are as often used as direct communication. An interesting side effect is that such cooperative systems can themselves then become a self-organizing system of systems.

An *autonomous* system performs the desired tasks and behaves well and adequate even in unstructured environments without continuous human guidance. In the domain of robotics, autonomy is described as a collection of additional characteristics, in particular sensor capabilities to observe chaotic, unpredicted variables and to react to keep the system on track utilizing the available degrees of freedom.

In general, *agile* systems are able to manage and apply knowledge effectively so that they behave well and adequate in continuously changing and unpredicted environments. In systems engineering, agility is often in particular connected with the development phase of systems and reflects the ability to immediately react on changes in the requirements.

Without doubt, additional characteristic properties can be identified that are desirable for such systems, such as self-healing. However, if a system is adaptive, elf-optimizing, and fault-tolerant, self-healing is a result. Similar arguments can be made for the quest to reduce risk and vulnerability and other desirable characteristics.

2.2 How to Capture Intelligence

There are many methods applied in AI to capture intelligence. This chapter deliberately focuses on a limited subset that is of particular interest to systems engineering and for which examples are given in other chapters of this books. Using the well known categories of Ackoff [2], we distinguish between data, information, knowledge, understanding, and wisdom. We understand *data* as a collection of facts. *Information* is data in a context allowing answering questions like who, what, where, and when. *Knowledge* is applied information answering the question how. *Understanding* introduces an answer to the question why, and *wisdom* finally evaluates understanding and generalizes the findings, allowing application of understanding in other domains than the original source of gaining understanding.

In this chapter and this book, we apply semantic systems or use general ontological means to capture and model data and information. Applying these pieces of information on who, what, where, and when in the context of simulation introduces the aspects addressed by knowledge: *how*. Adding agents allows running not only one but many simulations and comparing alternative courses of action. To communicate between agents, ontology is needed to provide the basis for the communication language supporting the exchange of information. Figure 2 shows the three elements applied in this book.

Fig. 2. Components to Capture Intelligence

A recent book edited by Yilmaz and Ören [3] copes with the various aspects of agent-directed simulation and systems engineering. They also show the increasing importance modeling and simulation methods in general and agent-directed simulations in particular play for intelligence-based systems. Software agents expose many of the characteristic properties described earlier in this chapter.

Agents help designing communication and coordination protocols in the system and may even become a surrogate for a human user. Simulation helps answering questions about the achieved behavior, performance and robustness, giving first feedback about the quality of the design. In addition, simulation can be used for decision support by providing "what if" scenarios as well as for training and education purposes. In addition, agents are likely to replace, to a certain degree, objects that have traditionally been exploited in systems engineering. An interesting aspect evaluated is to replace the functions traditionally developed within the functional architecture of a system as defined in [1] with agents. As this agent already possesses many characteristics of intelligence-based systems, the result is likely to be close to our objective. However, all three aspects shown in figure 2 are important.

Another example of interest described in [3] is autonomic computing, as it also shares many characteristic properties. Autonomic computing is a potential strategy and philosophy in systems design and management that aims to cope with increasing complexity in the presence of constant change addressing the area of systems of systems engineering which involves: (a) large scope and great complexity of integration efforts; (b) collaborative and dynamic engineering; (c) engineering under the condition of uncertainty; (d) continuing architectural reconfiguration; (e) simultaneous modeling and simulation of emergent behavior; and (f) stakeholders with competing goals and objectives.

Utilizing the characteristics of software agents, autonomic systems are based on architectures and mechanisms that facilitate self-configuration and adaptation through learning, anticipation, and robust designs to be able to adjust and fine tune system parameters to emerging situations in this environment. The main characteristics are self-configuration, self-healing, and self-optimization. The autonomic computing control loop moves from gathering data from resources in the system's environment (sensor) to registering to be notified as the sensors observe changes in the environment (monitor). Next, the status of the environment and operational components' ability to react to change is perceived, interpreted, and understood (analyze) while necessary information about the managed resources, data, and policies are being provided to the system (knowledge). If the analysis and knowledge cannot identify a proper reaction to unforeseen environmental conditions, the reasoning and planning components take control to generate a new plan and identify a sequence of actions to act on the system configurations. Then, those actions are translated into executable commands (execute). These key tenets of autonomic systems (sensor, monitor, analyze, knowledge, reason/plan, execute) provide a roadmap for building intelligence-based systems using all three components mentioned above.

However, the title of this book is not "systems engineering *of* intelligence-based systems," but "intelligence-based systems engineering," which also includes the application of these methods and technologies to improve the traditional systems engineering process and the emerging new field of system of systems engineering. The next section will describe the principles of systems engineering and identify where intelligence-based methods can be applied.

3 Systems Engineering

The genesis for systems engineering in particular in the United States has been attributed to complexity. Early pioneers in the systems engineering field emphasize increasing system complexity as the principal causative factor, although they recognize that this is far from a complete explanation [4] [5]. To explain this, some historical background is warranted.

In the late 1930s the fledgling radio, television, and telephone industries in the United States recognized the need for a systems approach in the development of modern telecommunications services. The Radio Corporation of America (RCA) and its subsidiary, the National Broadcasting Company (NBC) were interested in the expansion of their television broadcast domain. At the same time, the Bell Telephone Company was interested in the expansion of their long-distance telephone network. Both companies initiated technical studies aimed at increasing their markets through the use of new broadband technologies that were beginning to emerge in the early 1940s. However, these exploratory studies and experimentation were interrupted by the Second World War.

During the Second World War, the American military used large numbers of scientists and engineers to help solve complex logistical and strategic bombing problems related to the war effort. Many of these efforts made significant contributions to the philosophy and techniques of what was then called *Operations Research*. At the same time, the need for many novel types of electronic gear for airborne use gave rise to a wide variety of component devices, popularly known as *black boxes*. These were ingenious devices, but their application in terms of the entire system of which they were merely parts was a matter of improvisation [4]. Inevitably, many of the engineers and scientists working on these *black boxes* were required, by necessity, to look ahead to the ultimate goal – the system. When the war ended a number of corporations (most notably the RAND Corporation, the Bell Telephone Laboratories and RCA) hired much of this pool of talented scientists and engineers to provide services to both the government and the telecommunications industry. These seasoned practitioners were able to capitalize upon the lessons from their war-time experiences in the development and implementation of the modern telecommunications and electrical power systems. The telecommunications system development efforts provide for much of the early literature on systems engineering. Schlager [6], in a nationwide survey found that the Bell Telephone Laboratories was probably the first organization to use the term systems engineering. If true, this places the start of what we call systems engineering, in the early 1940s.

3.1 Traditional Systems Engineering

The emergence of systems engineering in the 1940s was an outgrowth of the need to deal with large, expensive systems. The early textbooks [5] [7] [8] on systems engineering had an emphasis on topics such as decision making, problem solving, and analysis of alternatives. The texts relied heavily on the techniques and analytical methods from Operations Research [9] [10] [11]. A 1957 definition of systems engineering characterizes its early role [12].

> *"The design of systems in which the output is a set of specifications suitable for constructing a real system out of hardware*." (p. 1-4)

Over the next 30 years systems engineering assumed responsibility for not only the technical elements surrounding systems, but the life cycle management responsibilities as well. Systems engineering was, in-part, responsible for the delivery of large complicated projects of national importance that included the Polaris submarine, and the Mercury and Gemini space programs. By 1998 the definition of systems engineering had evolved to [13].

> *"An interdisciplinary collaborative approach to derive, evolve, and verify a life-cycle balanced system solution which satisfies customer expectations and meets public acceptability."* (p. 11)

In 30 years, systems engineering had evolved to include life-cycle management responsibilities, customers, and the public in its definition. Traditional Systems Engineering (TSE) has developed the frameworks and methodologies [13] [14] to successfully conceive, design, acquire, and field large multi-purpose systems. Three often used models developed in support of TSE most readers will recognize are (a) the waterfall model [15], (b) the *Vee*-model [16], and (c) the spiral model [17].

The waterfall model is characterized by the sequential evolution of phases in which as a rule only the two consecutive phases are connected with each other and the feedback is seen as the exception, not the rule. It starts with a set of requirements that are refined for the system and its component, followed by an analysis. The analysis is followed by the detailed design and the implementation of this design. Once the system is implemented, it is tested and afterwards operationally used. Some newer versions include maintenance and retirement as well. All versions of the waterfall model have the philosophy in common that if the engineer is doing a good job in all phases, he can successfully reach the project end. The *Vee*-model follows a slightly different philosophy by integrating the user into the engineering process. It starts with user requirements and ends with user acceptance. The two parts of the *Vee* are built by the phases comprised in the *decomposition and definition* of system components in the downward steps, and the *integration and verification* phases building the upward steps. On all levels, phases of the decomposition and definition are connected to respective integration and verification phases, such as verifying that the correct parts are built, verifying that configuration items are assembled correctly, verifying that the system performs as requested, and validating that the system fulfills all requirements. Overall, the feedback between the different phases and the possibility of corrections of earlier phases that are not necessarily mistakes of the systems engineer build the philosophy. The last model, the spiral model, is based on waterfall and *Vee* model ideas. It assumes that several iterations through the phases of these models will be needed resulting in a spiral in which each iteration leads to the next iterations objectives. Feedback is the rule and no longer the exception. The *spiral* model is an iterative model that combines elements of the classic *waterfall* model with the characteristic of prototyping and produces an evolutionary approach to engineering. The four major phases are (1) management planning, (2) engineering, (3) customer evaluation, and (4) risk analysis. The major distinguishing feature of the spiral is that by including a formal risk analysis phases, it introduced a risk-driven approach to the development process.

Systems Engineering therefore evolved well. However, the 21st century presented a new problem for systems engineering: the system of systems. The next section will exemplify that the traditional methods are insufficient to address these new challenges so that a new theory is needed that allows to derive methods and implement solutions.

3.2 System of Systems

Most 20th century systems were designed and implemented to satisfy specific functional objectives. The objectives were typically focused on the requirements in a single functional area (i.e. accounting, inventory control, manufacturing, railroads, highways, etc.), resulting in a number of vertically independent, or stove-piped, systems within an organization or society. Few were designed to satisfy all of the functions required by the organization or society they were serving and as such are classified as monolithic in structure.

Today, large numbers of 20th century systems operate within these functional stovepipes, providing functionality inside but not across the stovepipes. Initial efforts to bridge the functional stovepipes have focused on integrating 20th century systems through a series of system-to-system interfaces. However, 21st century managers are no longer satisfied with disparate systems lashed together with complex interfaces and data validation routines. Enterprise Resource Planning (ERP) systems were supposed to be the panacea for the business world, replacing stove-piped legacy systems with a single system encompassing all of a company's functional requirements. In 1998 it was estimated that businesses around the world were spending $10 billion per year [18] on enterprise systems and that figure probably doubles when you add in associated consulting expenses.

By the turn of the century, a new type of system, beyond that envisioned by the late Russell Ackoff in his paper *The Systems Revolution* [19], began to emerge. It is the super-system, the metasystem, the system-of-systems which is made up of components which are large-scale systems themselves. If we are to understand system-of-systems we must be able to differentiate them from the more common monolithic systems.

Although the term system-of-systems has no widely accepted definition, Maier notes that the notion is widespread and generally recognized [20]. The following distinguishing characteristics have been proposed [20] [21].

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 component systems, these component systems are capable of independently performing useful operations independently of one another.

2. *Managerial Independence of the Systems:* The component systems not only can operate independently, they generally do operate independently to achieve an intended purpose. The component systems are generally individually acquired and integrated and they maintain a continuing operational existence that is independent of the system of systems.

3. *Geographic Distribution:* Geographic dispersion of component systems is often large. Often, these systems can readily exchange only information and knowledge with one another, and not substantial quantities of physical mass or energy.

4. *Emergent Behavior:* The system-of-systems performs functions and carries out purposes that do not reside in any component system. These behaviors are emergent properties of the entire system-of-systems and not the behavior of any component system. The principal purposes supporting engineering of these systems are fulfilled by these emergent behaviors.

5. *Evolutionary Development:* A system-of-systems is never fully formed or complete. Development of these systems is evolutionary over time and with structure, function and purpose added, removed, and modified as experience with the system grows and evolves over time.

These distinguishing characteristics begin to place some degree of formality on the notion of system-of-systems, but something is missing. In order to go beyond the traditional perspective of a fully integrated system-of-systems which perfectly shares data in what we call hard interoperability, we must invoke a more systemic view. The ideal state for a system-of-systems requires what we will call *systemic interoperability*. Systemic interoperability is a holistic view of interoperability and requires compatibility in worldview and conceptual, contextual, and cultural interoperability, allowing the system-of-systems to act consistently with regard to purpose, function, and form. In other words, it is not sufficient to align the implementation details of the participating systems, but the underlying conceptualization and the assumptions and constraints need to be aligned as well. This is where System of Systems Engineering comes into play.

3.3 System of Systems Engineering

During the evolution of TSE, the educational texts [22] [23] [24] and curricula eliminated topics on the fundamental concepts and properties associated with systems and include few *soft* topics to encompass the rich context and human situations that real-world systems of systems engineering problems present.

Man-made systems of systems require a holistic, systemic understanding of both the technical problem and the contextual framework present in order to arrive at satisfactory solutions. A new set of methodologies and frameworks based upon formal systems principles are required. The new methodologies will also require new supporting methods, techniques, and tools.

The emerging discipline of System of Systems Engineering (SoSE) is attempting to address the problems associated with systems-of-systems. Because these problems are *messy* traditional methodologies of systems engineering are excluded from consideration in this context. Russell Ackoff coined the concept of a “mess” and “messes” [25]:

> *“Because messes are systems of problems, the sum of the optimal solutions to each component problem taken separately is not an optimal solution to the mess. The behavior of the mess depends more on how the solutions to its parts interact than on how they interact independently of each other. But the unit in OR is a problem, not a mess. Managers do not solve problems, they manage messes.”* (p. 100)

Keating et al. [26] cite three important problems that TSE is not prepared to address when facing a complex metasystem problem:

1. Has not been developed to address high levels of ambiguity and uncertainty in complex systems problems . . . it strains to adequately respond to ill-structured problems with constantly shifting requirements.
2. Does not completely ignore contextual influences on system problem formulation, analysis, and resolution, but places context in the background.
3. Is not prepared to deliver incomplete or partial solutions that include iterative design and implementation after deployment.

Keating et al [26] provisionally define SoSE as:

> *"The design, deployment, operation, and transformation of metasystems that must function as an integrated complex system to produce desirable results. These metasystems are themselves comprised of multiple autonomous embedded complex systems that can be diverse in technology, context, operation, geography, and conceptual frame."* (p. 23)

Multiple, autonomous, embedded, complex systems function as a single meta-system, or system-of-systems. This is possibly the most daunting task ever presented to systems engineers. It exists within a unique new context and will require an entirely new methodological problem solving approach.

From a programmatic and enterprise viewpoint, TSE emphasized a system-centric view with individually designed, developed, implemented and optimized solutions, which necessarily incorporated the danger of stovepipes and fragmentation. What is envisioned, however, is an integrated system approach in which each system provides capabilities in an easy and composable way to support the rapid reconfiguration. To support this vision, a methodology is needed that guides systems engineers in the new system of systems problem domain.

3.4 System of Systems Engineering Methodology

Currently, there is no widely accepted approach to conducting System of Systems Engineering (SoSE) efforts. However, there is recognition that approaches must address challenges of increasingly complex systems that must be conceived, built, operated, and evolved in a changed landscape marked by: (1) an exponential rise in the demand, accessibility, and proliferation of information, (2) increasing interdependence and demands for interoperability between systems that have previously been developed, tested, operated, and maintained in isolation, (3) missions and flow down requirements that are subject to rapid and potentially radical shifts due to policy, organizational, funding, or other factors beyond the technical aspects of the system, and (4) demands for the accelerated fielding of systems that are technically incomplete, but offer an improved alternative to what is presently available [27].

The SoSE Methodology [28] is a rigorous engineering analysis that invests heavily in the understanding and framing of the problem under study. By conducting a rigorous engineering analysis of the problem and its associated context, the SoSE Methodology minimizes the chance that a Type III error or solving the wrong problem precisely and efficiently [29] [30] may be committed early on in a SoSE analysis. It is important that the SoSE Methodology is not taken as a prescriptive approach to addressing complex SoSE problems. Instead, the SoSE Methodology must be taken as a guide, to be adapted to the particular circumstances that define its application. Otherwise, it will not serve its intended purpose: to provide a high level adaptable structure to guide rigorous exploration of complex systems problem situations.

The SoSE Methodology is intended to provoke rigorous analysis – resulting in the potential for alternative decision, action, and interpretations for evolving complex system of systems solutions. The SoSE Methodology is based in facilitating inquiry that is as much about thinking and framing of problems, their context, and managing emergent conditions as it is about taking decisive action. The SoSE Methodology was purposefully built and seeks to provoke higher levels of inquiry, systemic analysis, and advance understanding of seemingly intractable problems enroute to more robust solutions.

We position the SoSE Methodology to be consistent with Checkland's [31] perspective of a methodology, which suggests that a methodology provides a framework, more specific than philosophy, but more general than a detailed method or tool. Therefore, a systems-based methodology must provide a framework that can be elaborated to effectively guide action. There are several critical attributes for a methodology and these are consistent with the current state of evolution for the SoSE Methodology. These critical attributes are discussed in the next section.

There are several critical attributes in the SoSE Methodology that are consistent with the current state of evolution for SoSE. Although the listing is certainly not intended to be exhaustive, we offer these as insight to our thinking with respect to the characteristics that make the SoSE Methodology sustainable. The nine (9) critical attributes and how the SoSE Methodology satisfies these are presented in Table 1.

Table 1. Critical Attributes of the SoSE Methodology

Attribute	Explanation
Transportable	Must be capable of application across a spectrum of complex systems engineering problems and contexts. The appropriateness (applicability) of the methodology to a range of circumstances and system problem types must be clearly established as the central characteristic of transportability.
Theoretically and Philosophically Grounded	Must have a linkage to a theoretical body of knowledge as well as philosophical underpinnings that form the basis for the methodology and its application. The theoretical body of knowledge for the SoSE Methodology is systems theory.
Guide to Action	Must provide sufficient detail to frame appropriate actions and guide direction of efforts to implement the methodology. While not prescriptively defining how execution must be accomplished, the methodology must establish the high level what's that must be performed.
Significance	Must exhibit the holistic capacity to address multiple problem system domains, minimally including contextual, human, organizational, managerial, policy, technical, and political aspects of a SOSE problem.
Consistency	Must be capable of providing replicability of approach and results interpretation based on deployment of the methodology in similar contexts. The methodology is transparent, clearly delineating the details of the approach for design, analysis, and transformation of the SOS.
Adaptable	Must be capable of flexing and modifying the approach, configuration, execution, or expectations based on changing conditions or circumstances – remaining within the framework of the guidance provided by the methodology, but adapting as necessary to facilitate systemic inquiry.
Neutrality	Attempts to minimize and account for external influences in application and interpretation. A methodology must provide sufficient transparency in approach, execution, and interpretation such that biases, assumptions, and limitations are capable of being made explicit and challenged within the methodology application.
Multiple Utility	Supports a variety of applications with respect to complex SOS, including, new system design, existing system transformation, and assessment of existing complex SOS initiatives.
Rigorous	Must be capable of withstanding scrutiny with respect to: (1) identified linkage/basis in a body of theory and knowledge, (2) sufficient depth to demonstrate detailed grounding in relationship to systemic underpinnings, including the systems engineering discipline, and (3) capable of providing transparent results that are replicable with respect to results achieved and accountability for explicit logic used to draw conclusions/interpretations.

The foundations of the SoSE Methodology are found in two primary aspects, namely (a) the theoretical and philosophical foundations for systems and (b) the seven perspectives of an enabling methodology shown in figure 3.

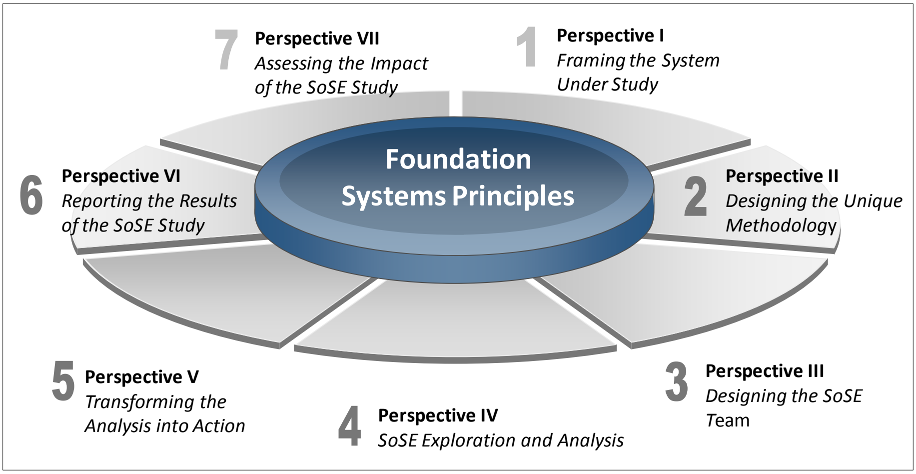

Fig. 3. The SoSE Methodology

First, the underlying theoretical and philosophical grounding are derived from systems theory. The principles, laws, and concepts central to the SoSE Methodology are from systems theory [32]. These principles, laws, and concepts are central to everything that follows in application of the SoSE Methodology to a specific problem domain. In effect, they define the thinking that supports following decision, action, and interpretation essential to effective SoSE. This sets the stage for a consistent approach to deployment of the SoSE Methodology by participants.

The second aspect of the SoSE Methodology is found in the seven perspectives that exist throughout a SoSE effort. Each perspective is:

- Essential to a holistic SoSE treatment of a problem area,
- Applied in iterative fashion throughout a SoSE project effort,
- Exists in relationship to all other perspectives, informing and informed by other perspectives,
- Can have a different priority at different times during an effort,
- Flexible in application, requiring tailoring depending on the context and problem domain, and
- Consists of detailed elements (that will vary in application) that serve to structure the application of the perspectives.

Each of the seven perspectives is briefly presented in the following paragraphs.

Perspective I: Framing the System under Study. This perspective is designed to rigorously structure the system problem, the contextual setting and environment within which the problem system exists. Key execution elements in this perspective include:

- Generalize the Wide Context for the System under Study – establish the circumstances, factors, conditions, and patterns that are characteristic of the situation surrounding the system of systems (SoS).
- Characterize the System under Study – understand the basic structure and characteristics of the system of systems under study, including the SoS's objectives, functions, environment, resources, components, and management.
- Characterize the Complex Nature of the System Domain under Study – establish the complex nature of the SoS and problem domain.
- Present the System Domain as Characteristically Complex - present the SoS under study as a complex systems problem.
- Frame the SoSE Problem - depict the problem situation by expressing the structure, elements of processes and the situation.
- Define Study Purpose, Reformulated Problem Statements and Objectives - clearly explain the nature, purpose, high-level approach, and objectives for the effort.
- Conduct Stakeholder Analysis - explicitly account for and address the multiple interests (rational and irrational, inside & outside) which can impact achievement of system objectives.
- Conduct Contextual Analysis - account for the set of circumstances, factors, conditions, values and/or patterns that are influential in constraining and enabling the SoS engineering process, the SoS solution/recommendation design, SoS solution/recommendation deployment considerations, and interpretation of outputs/outcomes stemming from the effort.

Perspective II: Designing the Unique Methodology. This perspective designs a unique methodology based on the problem and the problem context.

- Construct High-Level Design for the Study - construct a unique high-level methodology that will adequately support the study objectives and the SoS context. This must be compatible with the problem and problem context.
- Develop the Analytic Strategy - create the design for quantitative and qualitative exploration (data collection and analysis) necessary to understand and make decisions concerning the SoS under study.
- Develop Assessment Criteria and Plan - construct a set of measurable performance criteria that can be used during and after the problem study to ensure continued fit of problem, context, methodology and capability to meet study objectives.

Perspective III: Designing the SoSE Team. This perspective designs the team to undertake the SoSE study, taking into account the nature of the SoS problem and the team resources, skills, and knowledge that can be brought to bear for the problem.