Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 36 (2014) 41 – 48

Procedia
Computer Science

Complex Adaptive Systems, Publication 4
Cihan H. Dagli, Editor in Chief
Conference Organized by Missouri University of Science and Technology
2014- Philadelphia, PA

Quantitative SoS Architecture Modeling

Joseph W. Marvin, Robert K. Garrett, Jr.

1300 S. Litchfield Road, Goodyear, AZ 85338
Missile Defense Agency, Colorado Springs, CO 80912, USA

Abstract

Improved techniques to develop future System of Systems solution architectures are a systems engineering imperative. The systems engineer must be able to produce quantitative Measures of Performance that parse good performing architectures from bad architectures in a very complex and dynamic domain often and early during concept development. This paper proposes innovative System of Systems Engineering techniques to manage risks and reduce costs in complex systems. We suggest a combination of graph theory, Big Data and Uncertainty Quantification are foundational tools that can model architectures and more importantly the movement of data across architecture interfaces. This provides a mathematical foundation for quantitative architecture, model based systems engineering and simulation environments enabling continuous prototype testing in a model environment. The result is a means to quantitatively validate viable complex architectures early in the lifecycle. This paper describes a graph-based approach to model complex architectures. Complexity will be defined mathematically as a function of the number of entities, inter-entity relationships and active event sequences in the graphical architecture. Uncertainty quantification compares the graph model to the physical System of Systems based on quantities of interest. Big Data facilitates analytics across continuous data threads in the architecture.

Keywords: System of Systems; Architectures; Ontology; Mission Threads

1. Introduction

The 21st century will be defined by the human ability to solve complex system challenges. In these complex system challenges combinatoric possibilities in the solution space and continuously evolving requirements and environments are daunting. Extensibility of our current systems engineering toolbox is questionable. Space exploration beyond our solar system, control of global weather, genetic regeneration and future conflicts fought entirely by cyber robotics are examples. The systems engineer must be able to produce quantitative, operational, goal-based Measures of Performance (MoP) that parse good performing architectures from bad architectures in a

doi:10.1016/j.procs.2014.09.104

dynamic, very complex domain. This paper proposes System of Systems Engineering (SoSE) is best served by shifting to graph theory to focus on dynamic interactions and relationships. The unstructured infrastructure of Big Data analytics, and Uncertainty Quantification to establish tolerance bounds. This provides a mathematical foundation for Model Based Systems Engineering (MBSE) and simulation environments enabling continuous prototype testing in a synthetic environment. The result is a means to reduce complexity and quantitatively validate viable architectures early in the lifecycle.

While many component technologies may exist today, techniques to effectively and dynamically integrate these technologies to meet operational goals are missing. This work will leverage graph-based technologies and the complex SoS will be modeled as a graph [Garrett, 2011 and 2014]. Complexity will be defined mathematically as a function of the number of entities, i.e., systems, sub-systems, inter-entity relationships and active event sequence or scenarios [Bonchev, 2005]. Uncertainty quantification [Marvin, 2014; DeLaurentis, 2000] plays a fundamental role to characterize the impact of variability (aleatoric) and lack-of-knowledge (epistemic) variables of the graph model as compared to the physical SoS based on quantities of interest. Big data is the process of managing a dynamic data store and the use of inference techniques for data interrogation and analysis [Shvachko, 2010] across the complex systems and SoS lifecycle focused on relationships. These relationships, the edges of the graph are the essence of interfaces and integration [Garrett, 2011]. We present the concept of using graph theory to parse the graph using inference techniques and to transform class-based, hierarchical set of linked nodes, and a classic ontology [Graves, 2008], into operational data threads across the architecture thus reducing both complexity and uncertainty. This construct allows direct early modeling of data threads across diverse component systems within the SoS, or localized specific views within the enterprise to provide constraint and structure for subsequent architecture development [Piatti, 2010]. Uncertainty quantification techniques add powerful forensics that bring static architectures to life and allows architecture comparisons and synthesis of the best components of many architectural options.

1.1. SoSE Quantitative Architecture Modeling

Quantitative architecture evaluation is the only means to evaluate characteristics such as persistence, resilience, flexibility and responsiveness. Graph theory is not new. The concept of graphs has been around since Eulers Seven Bridges of Königsberg problem in 1735. Graphs eliminate the limitation found in the hierarchical structure used by architects and architecture tools. Social media (Google, Facebook, Twitter) (Malewicz, 2010; Google, 2013; Facebook, 2013; Russell, 2013) and cyber physical security have recognized this limitation and shifted to graph theory based approaches that lend themselves to a much richer analytical environment capable of tracing digital data threads through the system or architecture. Graphs that are contiguous in time are the only way to trace the data through collection and allow the numerical analysis necessary to evaluate complex SoS architecture operations.

Figure 1 provides a process description of typical architecture development based on the authors' experience. Traditionally, architecture development is conducted at the beginning of the lifecycle. In this model developmental testing is usually not conducted until after design activities near completion and significant cost and effort expended in manufacturing infrastructure. Test failures are frequent and the point of origin for the failure(s) is rarely pinpointed with subsequent design changes based not on engineering rigor but on previous investment decisions. Too often these architecture artifacts tend to be static views that have limited utility later in the lifecycle (oftentimes filed away and later updated to the "as built" architecture. Major System Acquisition procurement contracts initially focus on requirements at the expense of architectural and employment concepts. We think a shift is necessary for complex SoS developments. The shift changes early lifecycle SoSE to a math based architecture effort that invests quantifiable architecture development. SoS procurement strategy could change from requirements to mission threads.

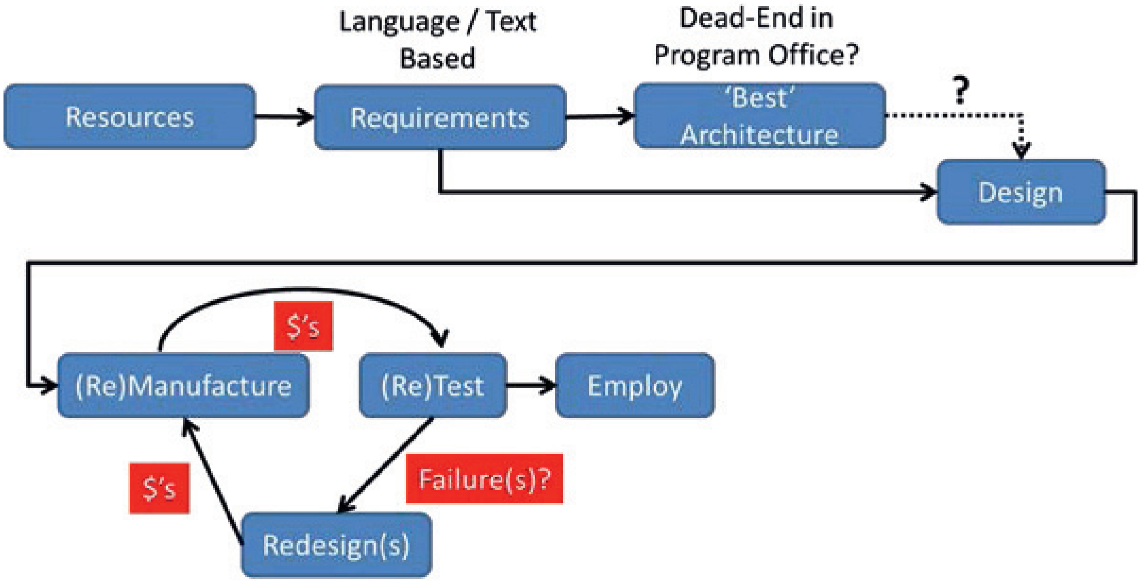

Figure 1. Current use of architecture in a development process

1.2. Graph Based Architecture

The question is, "How can we evaluate architecture concepts against mission capabilities of interest?" The answer is that this is not possible with the static, language based, architecture development methods currently in use by systems engineers today. Our graph based approach proposes a process shift to math based rigor that can address this deficiency in current systems engineering practices. The work described in this paper includes the research to develop a quantifiable architecture development process. A fundamental question to be addressed in the development of a complex system or SoS architecture given the significant number of plausible combinatoric possibilities is which architectural bases are good enough to meet mission goals. We suggest the process shown in Figure 2 is the only way to quantitatively answer questions posed in today's architecture developments. Our concept applies mathematical architecture techniques that focus on the mission environment and goal-based mission threads to quantitatively answer key questions about architecture alternatives. Architectures are evaluated early and often in run time environments providing for an understanding of break points and performance boundaries. Our architecture process addresses risk issues that have repeatedly challenged complex SoS architectures. Risk will be identified as patterns in the architecture bounded by uncertainty. High uncertainty is equated to high risk. All areas of a common pattern can be readily identified and tracked as a risk, and monitored through continuous testing.

We have developed a quantitative architecture concept by leveraging existing architecture fundamentals such as DoDAF, SysML, and ontology. These can all be poised on a graph basis and are consistent with the super ontology, mission environment and mission thread constructs. What we have recognized is that dynamic, quantitative methods have been missing from these systems engineering tools. Our research finds that by addition of very basic constructs, such as graphs and graph theory, a new dimension of quantitative analysis techniques become available to the systems engineer. We suggest that the systems engineering domain must embrace quantitative math based techniques and rigor to effectively address the SoS challenges of the future.

Our concept starts by considering requirements, resources, operations and policy within the context of mission goals. Conceptually, as we proceed time is dedicated to iterative building and testing the ontology. This provides us the common basis for definitions and a lexicon to collaborate among stakeholders. We can now begin to develop concepts, and entity relationships that can be unambiguously represented in graphs using graph theory. These relationships are developed as mission threads that represent the combination of mission goals in the operational environment necessary to perform the mission objectives of the architecture under evaluation. Graphs are executable using inference engines testing for consistency and composability in a variety of open-source tools such as Gephi, NetworkX and represent run-time models of a digital data thread through an architecture. Aggregated together, these digital data mission threads represent a Run-Time environment of a specific architecture.

We observe that many major system acquisitions run into programmatic troubles well beyond major milestones supported by well-intentioned systems engineers. Why has systems engineering so often failed? Complex systems when tested in a SoS context or when employed as part of an SoS have not met expectations in the up-front, requirements driven processes that have worked in previous system developments. The current state processes don’t provide the necessary context, structure and constraint given all the combinatoric possibilities in plausible employment concepts and the dynamic pace of change in the mission environment. The lessons learned continue to tell us that it is the interfaces, interactions and software driven data movement across these interfaces for a given operational goal in a specific employment configuration that drive programs red. Our new architecture method shown in Figure 2 emphasizes graph theory, Big Data and Uncertainty Quantification techniques. We do this by creating a mission environment (ME). Initially the ME is an ontology, either OWL-based or as a SysML Block Definition Diagram [OWL2, 2012; SysML, 2013], comprised of legacy knowledge from the stakeholders, the resources, i.e., the legacy entity systems / sub-systems, operational doctrine, and constraining policy. This class-based, hierarchical ontology can be tested using inference techniques based on graph mathematics such as centrality, density, clustering and consistency in language. The mission threads are based on a specific goal and operation and are a plausible sequence of events to achieve the goal. These mission threads can be viewed as plausible UML [UML, 2013] or SysML sequence diagram. The digital mission thread models are based on State Machines and become a testable simulation. Through the evolution of the mission thread models and subsequent simulations and testing, patterns of goodness can be established ultimately becoming an architecture. Logically this provides the continuity across the architecture and the application of Big Data analytics and emerging parametric and statistical analysis techniques become useable. This basis pushes iterative analysis of many architecture alternatives early in the lifecycle, focuses on goal-based mission threads and early risk reduction of the areas that will cause costly problems later in the lifecycle.

In this new approach, the focus is on the area that matters the most, the mission environment. Our investment is in the ontology, mission goals, and development of mission threads represented as executable mission models and ultimately mission simulations. The outputs of these simulations are Sequence Diagrams and supporting performance data.

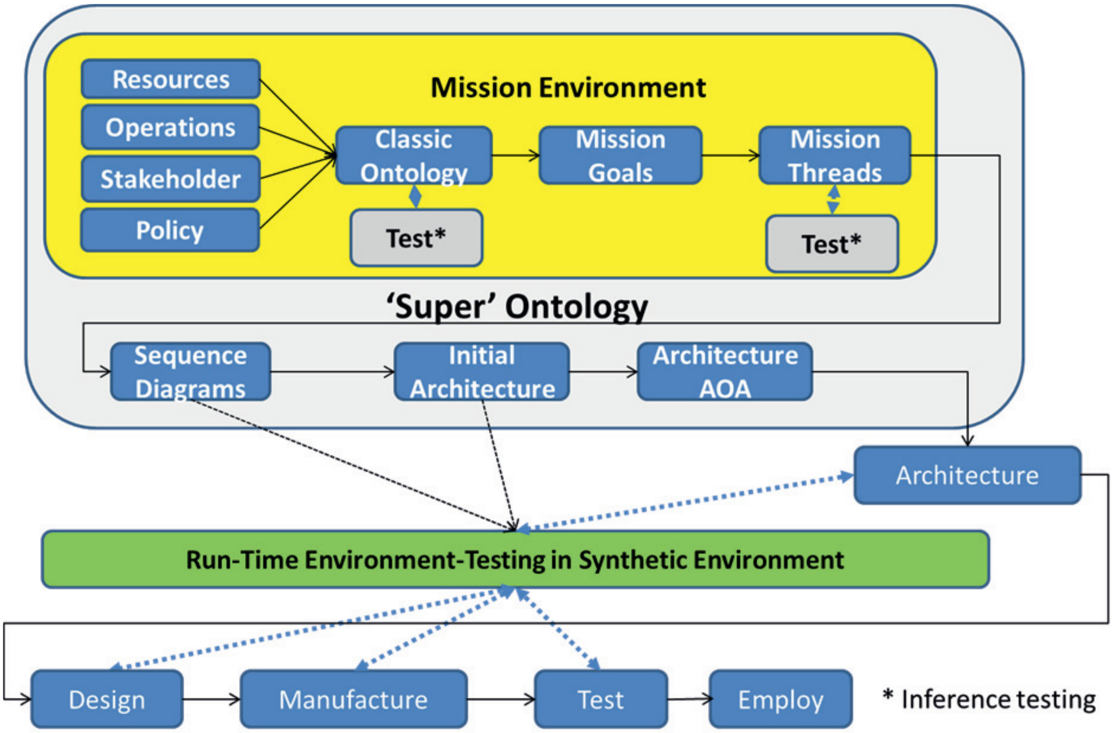

Figure 2. Proposed SoSE process with an expanded, dynamic and testable architecture

The basis of this approach is the use of existing graph-based MBSE tools, i.e., UML and SysML which provide an appropriate level of tested context, structure and constraint. These tools can also produce executable models leading to simulations where verification and validation back through the graph-based artifacts is achievable. Out of this early work comes a "Super Ontology" model that facilitates collaboration and initial architectures. The shift to graph theory allows us to inject testing of mission threads via mission model simulations in the mission context early and often. Not until there is a combination of mission threads do we begin architecture development driven by quantifiable testing via simulation of the plausible and subsequent parsing to bounding conditions. An architecture built in graphs can serve the entire lifecycle of design, manufacturing, testing and employment.

1.3. Data Analytics

Big Data analytic software tools can now be applied to literally hundreds and thousands of simulation runs to learn about system behavior based on real data moving through the objective architecture. We anticipate data analytics will play an important role as theoretical foundations for quantitative systems engineering are established [INCOSE 2014]. Our observation is that the architecture development methodology we are proposing will quickly become a big data challenge that will require a new family of enabling information technology tools and techniques that put engineering analytics into architecture SoSE.

Using data analytics, we can evaluate sequence diagrams, data aggregation, flows and model information about how the architecture will perform operationally. The literature describes challenges managing graph data, mining graph data and graph applications [Wang, 2010]. These are the challenges we want to embrace in the transformation of systems engineering into quantitative architecture modeling. New algorithms that can accommodate the dynamic and time-evolving nature of architecture graphs support our proposed transformation. These graph data analytics are influenced by the "super ontology" shown in Figure 2 above and find applications in the run-time environment as well as post data analysis.

The data analytics environment needed also relies on advances in high performance computing suggesting the need to consider this dimension in future SoSE. Techniques such as HADOOP Line Graph Fixed Point (HA-LFP) are capable of implementing techniques such as Belief Propagation (BP) and Inference [Kang]. These techniques provide for efficiency, scalability and effectiveness in line graph and directed graph applications. These tools add new dimensions to architecture methods which enable threat and boundary analysis on billion node architecture graphs. We can conceptually test how these architectures will perform without having written the first line of development software.

1.4. Uncertainty Quantification

Graphs and data analytics provide the contiguous and numerical foundation to apply robust statistical and parametric techniques providing for uncertainty tolerance bounds in a given performance space. The author's study [Marvin, 2014] of uncertainty and uncertainty quantification in complex SoS M&S environments motivated the use of graphs as the means for providing the ability to map mission threads across architecture components or entire SoS architecture models. This mapping of data provides a framework for the proper mathematical treatment of epistemic and aleatoric probabilities to demonstrate the effectiveness of quantifying and characterizing the output uncertainty of complex M&S arising from uncertainty in the input parameters. The technical feasibility of applying UQ techniques to M&S environments was demonstrated in a Scientific Technology Transfer project for the MDA in 2013. That demonstration showed the following:

- A consistent treatment of parameters with either probabilistic or possibilistic uncertainty without discarding evidence-based probability information or introducing spurious probability distributions;
- A biologically-inspired exploited search procedure to find areas of noteworthy performance; and
- An evidence-based fuzzy set of plausible outcomes derived from the mathematics of upper probability.

During this work, we identified the potential of UQ and other analytic tools applied to SoS architecture modeling. Our topical UQ approach provides an analytic capability to evaluate executable models during development at component and enterprise levels and can be used to characterize architecture performance boundaries and break points. We anticipate the architecture development process we have described will leverage a variety of tools in addition to UQ. Our near term work is applying UQ to architecture modeling and we will present its utility in future publications.

1.5. Summary

We have described an end-to-end process for Quantifiable Architecture development. As of this writing our research has accomplished the following:

- Suggested the use of graph theory as a means for more accurately modeling and testing of constituent systems and constituent systems within a SoS providing context, structure and constraint for the use of UML and SysML tools. UML and SysML are powerful graph-based tools that can readily exploit existing open source graph tools.
- Interacted with IBM Worldwide Big Data Team to learn their progress and applications consistent with the use of unstructured data, graphs, and data analytics (reported in this paper).
- Put forward the notion of uncertainty characterization and quantification as math based analysis tool enabled by graph and big data techniques.

A Quantifiable SoSE architecture development process is summarized in Figure 3. In the process, testable architectures are developed and tested prior engineering design. The process involves the creation of a dynamic data model of the SoS domain intended to be a knowledge management basis over the SoS lifecycle. The process starts with the collection and organization of legacy knowledge about requirements, resources, operations and policy into a class-based, classic ontology. Employment-based mission goals are then established for the SoS. These goals along with the associated operational doctrine provide the basis to establish plausible mission threads. There are many plausible mission threads for each set of goals and operations due to the SoS complexity and combinatoric possibilities. Through inference testing, a plausible set of ‘good enough mission threads can be established. Based on a functional decomposition these mission threads can readily be transposed into UML or SysML to produce mission models and simulations the output of which are Sequence Diagrams. The simulations can be initially based on Bayesian statistics, Marchov analyses, and/or creedal networks to quantify probabilities of success before detailed parametric and tactical algorithms are developed. The output of this testing is a set of initial SysML based concept architectures to further refinement and detail using standard MBSE tools. Another feature of the process is the establishment of two separate architectures from a common model basis; one for the Tactical Environment, and one for a Run-Time Environment. The run-time environment is the synthetic testing domain and is comprised of various Live, Virtual and/or Constructive simulation capabilities [M&S, 1998].

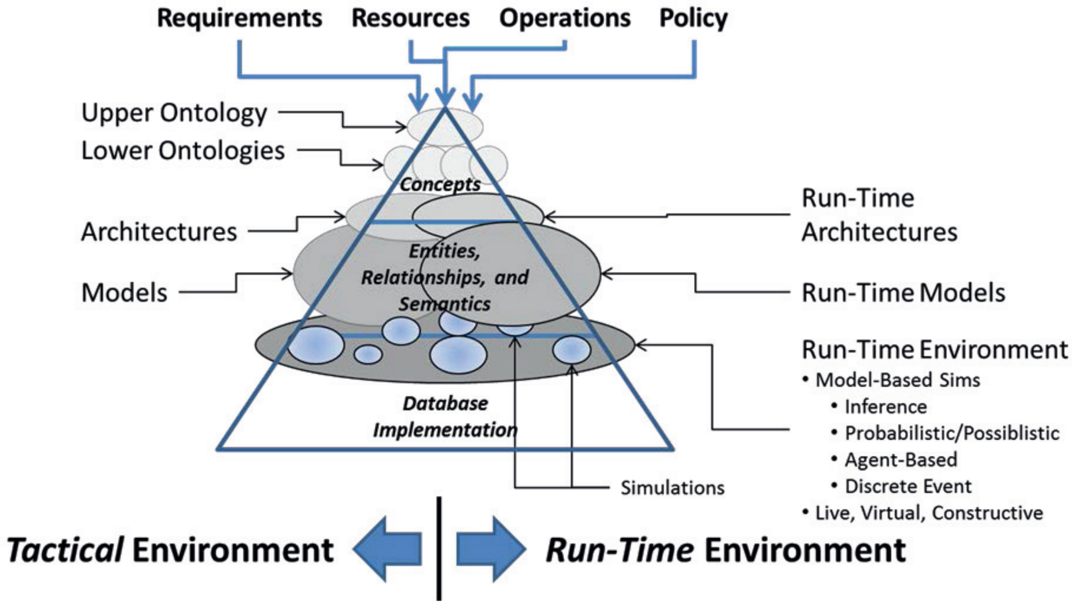

Figure 3. A SoSE process where a tactical environment and a run-time environment are created from a common model basis

Quantitative architecture concept can result in a fundamental shift in acquisition strategies from requirements based acquisition to mission thread performance acquisitions. To make this point, refer back to Figure 2. Here we suggest a Future State in developing a SoS Architecture. In this future state, SoSE can overlay the DoD Acquisition model in a framework that provides testable results well ahead of current methods. For example, the first test block could be graph based inference testing to support Major System Acquisition (MSA) Milestone A. Mission Goals graphed as Mission Threads can be tested in the next block as a MSA Milestone B. Proceeding along the sequence in Figure 1, a Mission Goal approach does not yield an architecture until much later in the sequence than current methodologies – and after exhaustive graph based data analytics and testing of perhaps hundreds of alternatives. This is a logical place for a Preliminary Design Review (PDR). Imagine the expanded decision space at PDR aided by validated graph based mission performance threads. Finally, the Design block is the natural stage for the Critical Design Review (CDR). This milestone decision gate is informed by rigorous testing of hundreds of architecture combinations, Big Data analytics over terabytes of mission performance data and iterative UQ techniques. This is the necessary information for building resilient future architectures consistent with the INCOSE SE Vision 2025 [INCOSE, 2014].

One final note. Our search for a quantitative SoS architecture modeling process has come with serendipity. Graph theory used to describe complex mission environments took us on a quest for a modeling capability that enables quantifiable math is an achievable framework. The pleasant surprise has come with the recognition that the necessary framework is available through SysML, UML and data analytics. In addition, these tools and systems thinking are described as Core Elements of SoSE [OUSD, 2008]. What is left is the follow through commitment to put engineering back into systems engineering.

References:

Bonchev, D., & Buck, G. A. (2005). Quantitative measures of network complexity. In Complexity in Chemistry, Biology, and Ecology (pp. 191-235). Springer US.DeLaurentis, D. A., & Mavris, D. N. (2000). Uncertainty modeling and management in multidisciplinary analysis and synthesis. AIAA paper, 422, 5.

Garrett, R.K., Jr., 2014. A Graph-Based Metamodel for Enabling System of Systems Engineering. Submitted for consideration to Systems Engineering.

Garrett, R. K., Anderson, S., Baron, N. T., & Moreland, J. D. (2011). Managing the interstitials, a system of systems framework suited for the ballistic missile defense system. Systems Engineering, 14(1), 87-109.

Graves, H., & Horrocks, I. (2008, April). Application of OWL 1.1 to Systems Engineering. OWL Experiences and Directions April Workshop.

Marvin, J., Whalen, T., Morantz , T., Deiotte, R., Garrett, R.K., Jr., (2014). Uncertainty Quantification (UQ) in Complex System of Systems (SoS) Modeling and Simulation (M&S) Environments, INCOSE International Symposium, 2014

M&S (2009). "DoD Modeling and Simulation (M&S) Glossary", DoD 5000.59-M, DoD, January 1998. Retrieved from http://www.dtic.mil/whs/directives/corres/pdf/500059m.pdf.

Shvachko, K., Kuang, H., Radia, S., & Chansler, R. (2010, May). The hadoop distributed file system. In Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on (pp. 1-10). IEEE.

SySML (2013). OMG Systems Modeling Language. Retrieved from http://www.omgsysml.org/

OWL 2 Web Ontology Language Document Overview (Second Edition), 2012. Retrieved from http://www.w3.org/TR/owl2-overview/

Piatti, A., Antonucci, A., & Zaffalon, M. (2010). Building knowledge-based systems by credal networks: a tutorial. Advances in Mathematics Research, 11.

UML, 2013. OMG Unified Modeling Language. Retrieved from http://www.uml.org/

Malewicz, G., Austern, M. H., Bik, A. J., Dehnert, J. C., Horn, I., Leiser, N., & Czajkowski, G. (2010, June). Pregel: a system for large-scale graph processing. In Proceedings of the 2010 ACM SIGMOD International Conference on Management of data (pp. 135-146). ACM.

Facebook (2013). Facebook Open Graph. Retrieved from https://developers.facebook.com/docs/opengraph/

Google, 2013. Google Knowledge Graph. Retrieved from http://www.google.com/insidesearch/features/search/knowledge.html

Russell, M (2013). Twitter Interest Graph. Retrieved from http://programming.oreilly.com/2013/10/investigating-the-twitter-interest-graph.html

A World in Motion – Systems Engineering Vision 2025, Copyright ©2014 by INCOSE

C.C. Aggarwal and H. Wang (eds.), Managing and Mining Graph Data,
Advances in Database Systems 40, DOI 10.1007/978-1-4419-6045-0_1.

U Kang, Duen Horng Chau, Christos Faloutsos. Mining Large Graphs: Algorithms, Inference, and Discoveries, IEEE International Conference on Data Engineering (ICDE) 2011, Hannover, Germany

Director, Systems and Software Engineering, Deputy Under Secretary of Defense (Acquisition and Technology) Office of the Under Secretary of Defense (Acquisition, Technology and Logistics) Systems Engineering Guide for Systems of Systems, Version 1.0, 2008