Adaptive Architecture in the Ballistic Missile Defense System of Systems

1 Introduction

In a System of Systems (SoS), an architecture defines how systems interact through non-physical communication [1], which impacts SoS performance. The ability to adapt between various architectures might allow for an SoS to optimize its performance for a given situation. In the Ballistic Missile Defense System (BMDS), system interaction is handle by Command and Control, Battle Management, and Communications (C2BMC) [2]. Under Maier's architecting principles, C2BMC would classify the BMDS as a directed architecture, in which individual systems are able to operate independently, but are centrally controlled by a master decision maker [1]. However, if the BMDS were to experience network outages or be the target of cyber warfare, a decentralized acknowledged architecture may prove advantageous in reaction to the new operating environment. This research aims to identify possible scenarios in which an alternate architecture would benefit BMDS performance and propose a method to optimize the BMDS architecture as a situation develops.

2 Background

Adaptability of an SoS has largely been focused on the evolution of an SoS over time when acquiring new systems, not actively modifying the architecture in response to a changing environment. However, many of the principles applied to the evolution of an SoS over time still provide value and insight into adaptability if the BMDS is framed as an acknowledged SoS.

An acknowledged SoS is characterized as a System of Systems sharing common objectives, management, and authority, but systems still retain their own management and authority in unison with the overarching SoS goals [3–5]. There has been significant effort to characterize such systems and their evolution over time, as well as proposed methods to better optimize how such systems can negotiate and plan for optimal capability. (TODO here: Dynamic optimization, trade contracts, MUSTDO, etc). Regardless of the method, a fully realized plan for the evolution of a system requires that systems with authority acknowledge possible risks and plan for mitigating those risks [5].

In contrast to the wealth of literature on managing evolutions of an SoS on a large time scale, relatively few papers have been published investigating how an existing SoS may adapt itself to better fit a changing environment. Adaptability is characterized as the ability of an SoS to react when environmental factors reduce the SoS's performance, thereby regaining some of the lost performance [6]. In contrast, evolution of an SoS takes place on long time scales with deliberate action from the SoS authorities in order to increase capability in its expected environment [].

Some system-level approaches exhibit similar problems to an SoS. For example, characterizing the performance of alternate configurations of subsystems mirrors many challenges faced by characterizing the performance of different SoS architectures. Optimization through adaptation of a system simply requires the ability to adjust component parameters [7]. Haris et. al. proposed a method for using a combination of fuzzy logic, quality function deployment, and genetic algorithms to characterize the available performance space of subsystems in relation to customer goals, risks, and budget [8]. While such approaches do show success in analyzing alternative configurations of subsystems, they fail to address critical properties that separate an SoS from a simple system. Systems-level approaches characterize a single assemblage of components which act in unison to perform a function, whereas an SoS is an assemblage of systems that share a common goal [1]. In consequence, systems-level adaptivity or optimization approaches do not address how components can be reconfigured or reorganized in order to improve performance, but rather target the modification of component parameters or the selection of a subsystem.

3 Objectives

1. Characterize the costs and risks associated with switching between architectures

2. Identify when and where adaptive architectures in the BMDS may be useful

3. Create a notional test scenario and train a SODA model to act as a surrogate predictor of system performance under different architectures

4. Identify performance thresholds in which alternative architectures are more advantageous
5. Demonstrate improved BMDS performance

4 Cost and Risk

References Cited

[1] M. W. Maier, "Architecting Principles for Systems-of-Systems," *INCOSE International Symposium*, vol. 6, no. 1, pp. 565–573, 1996.

[2] "Command and control, battle management, and communications (c2bmc)."

[3] S. Agarwal, L. E. Pape, N. Kilicay-Ergin, and C. H. Dagli, "Multi-agent based architecture for acknowledged system of systems," *Procedia Computer Science*, vol. 28, pp. 1–10, 2014.

[4] S. Agarwal, "Computational intelligence based complex adaptive system-of-systems architecture evolution strategy," 2015.

[5] D. N. D. D. A. Fang, Z., "Multi-Stakeholder Dynamic Optimization for Acknowledged System-of-Systems Architecture Selection," *IEEE Systems Journal*, 2018. Accepted for Publication 2018.

[6] R. L. Ackoff, "Towards a System of Systems Concepts," 1971.

[7] a. de Roo, H. Sozer, and M. Aksit, "An architectural style for optimizing system qualities in adaptive embedded systems using Multi-Objective Optimization," *Joint Working IEEE/IFIP Conference on Software Architecture, 2009 European Conference on Software Architecture. WICSA/ECSA 2009*, pp. 349–352, 2009.

[8] K. Haris and C. H. Dagli, "Adaptive reconfiguration of complex system architecture," *Procedia Computer Science*, vol. 6, pp. 147–152, 2011.