Modeling System of Systems Acquisition

Nil Kilicay-Ergin
Great Valley School of Graduate Professional Students
Penn State University
PA, USA
nhe2@psu.edu

Paulette Acheson
Engineering Management & Systems Engineering Department
Missouri University of Science & Technology,
MO, USA
pbatk5@mail.mst.edu

John Colombi
Air Force Center for Systems Engineering
Air Force Institute of Technology
OH, USA
john.colombi@afit.edu

Cihan H. Dagli
Engineering Management & Systems Engineering Department
Missouri University of Science & Technology,
MO, USA
dagli@mst.edu

Abstract – *System of systems (SoS) acquisition is a dynamic process of integrating independent systems. This paper describes modeling of the SoS acquisition environment based on the Wave Process Model. Agent-based modeling methodology is utilized to abstract behavioral aspects of the acquisition process.*

Keywords: system of systems, system of systems acquisition process models, agent-based modeling

1 Introduction

Today, increasing number of system acquisitions focus on integration of independent systems into a System of Systems (SoS). Traditional system engineering approach of scoping the system boundaries and optimally allocating requirements to system components is not suitable for SoS development anymore. This is mainly due to the fact that SoS component systems are independent and have their own functionality, development processes, funding and operational missions. In addition changes in external environment such as funding, national priorities can alter the dynamics of the acquisition process [1], [3]. Therefore, SoS engineering processes need to consider change as an important dynamic beyond technical considerations.

Evolutionary acquisition models are more suitable to SoS development as they emphasize stakeholder involvement, interim milestones, increased iteration and concurrent development [4]. The Systems Engineering of SoS (SoS SE) model is developed by the US Department of Defense to identify the core elements of the SoS acquisition [1]. The Incremental Commitment Model [3] is a risk driven framework that can be tailored for SoS development. The Wave Model [1] maps SoS SE model's core elements to a series of time-sequenced iterative process to guide implementation of the framework for practitioners of SoS development.

Regardless of the acquisition framework, in order to address SoS needs, SoS architects have to collaborate with individual system architects to leverage individual system functionalities. However, most system architects assume that SoS participants exhibit nominal behavior but deviation from nominal motivation leads to complications and disturbances in systems behavior. It is necessary to capture the behavioral dimension of SoS architecture to be able to represent the full problem space to guide SoS analysis and architecting phase [5].

This paper builds on the Wave Process Model to abstract behavioral aspects of the acquisition process. An agent-based model of the process is discussed to analyze the impact of individual system motivations on the overall SoS architecture evolution. It is envisioned that this type of model will help us in understanding the intricate dynamics of the SoS development and improve acquisition process. In the following sections, the Wave Process Model is provided as background information (Section 2), a conceptual agent-based model of the process is described (Section 3), and finally future directions for the research are discussed in the conclusions section.

2 SoS SE: The Wave Model

The Wave Process Model presents the core elements of SoS SE model in a series of six time-sequenced major steps. The wave model or bus-stop approach is a development approach that is similar to the effect of periodic waves crashing at the shore or a bus that periodically stops at a specific location. The SoS has specific places in the development where it can accept updates from the individual systems. Individual systems can plan their deliveries to coincide with the SoS 'bus-stops' or can evaluate the effect of missing a planned SoS wave. Figure 1 illustrates the major elements of the Wave Process Model. The steps in the model are briefly introduced below as background information. For further details refer to [1].

2.1 Initiate SoS

This step involves understanding the SoS objectives and operational concept (CONOPS) as well as gathering information on core systems to support desired capabilities.

2.2 Conduct SoS Analysis

This step establishes an initial SoS baseline architecture for SoS engineering based on SoS requirements space, performance measures, and relevant planning elements.

2.3 Develop and Evolve SoS Architecture

This step evolves the initial SoS baseline and develops the SoS architecture. The SoS architecture includes individual systems, key SoS functions and interdependencies among systems. The architecture identifies necessary changes in contributing systems in terms of interfaces and functionality in order to implement the SoS architecture.

2.4 Plan SoS Update

This step plans for the next SoS upgrade cycle based on the changes in external environment, SoS priorities, options and backlogs.

2.5 Implement SoS Update

This step establishes a new SoS baseline based on SoS level testing and system level implementation. This step is the end of wave cycle or 'bus-stop' where updates from individual systems can be integrated into the SoS.

2.6 Continue SoS Analysis

This step is the beginning of the next wave cycle and continuous to analyze the current SoS architecture for future SoS evolution.

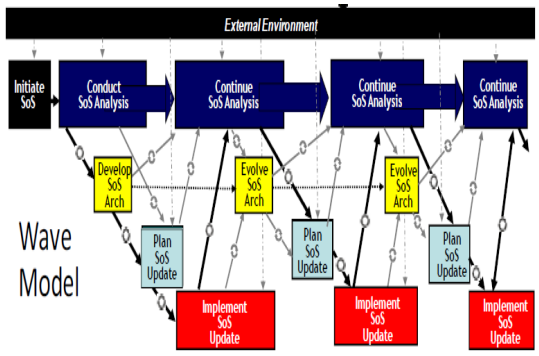

Figure 1. SoS Wave Process Model [1]

3 Agent-based SoS Acquisition Model

Agent based models (ABM) consist of a set of abstracted entities referred to as agents, and a framework for simulating agent decisions and interactions. Agents have their own goals and are capable of perceiving changes in the environment. System behavior (global behavior) emerges from the decisions and interactions of the agents. The approach provides insight into complex interdependent processes. Agent-based modeling methodology has several benefits over other modeling techniques; it captures emergent patterns of system behavior, provides a natural description of a system composed of behavioral entities and is flexible for tuning the complexity of the entities [6]. The methodology is used in a wide range of application domains such as financial markets [8], autonomous robots [9], and homeland security [10] to analyze and understand behavior of complex systems.

The agent-based methodology is suitable for analyzing the behavioral aspects of the acquisition process as agents can capture independent systems' behavior and SoS engineering activities. Figure 2 illustrates the agent-based SoS acquisition model. In the model, agents represent individual systems where agents embody the systems and the people responsible for them. The Wave process model applies to acknowledged SoS [7] thus there is also a specific agent responsible for the SoS engineering effort and for coordinating the individual system agents.

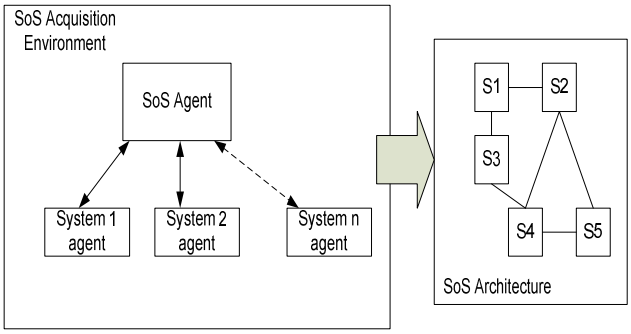

Figure 2. Agent-based SoS Acquisition Model

The following subsections outline the general structure of the agent-based model and how key elements of the Wave Process model are abstracted for analyzing the impact of individual system behavior on SoS architecture and acquisition process.

3.1 SoS Acquisition Environment

The SoS agent and the individual system agents are influenced by the changes in the SoS acquisition environment. Thus the environment model includes external factors/variables such as national priorities, threats and SoS funding. As the SoS acquisition progresses through wave cycles, these variables are updated to reflect acquisition environment changes. Table 1 summarizes the model elements in mathematical notation.

Table 1: SoS Acquisition Environment

External factors/variables:
$E_0 = f(National\ priorities, SoS\ funding, threats)$
Changes in external environment at wave time T: σ_T
External factors/variables at time T: $E_T = E_0\sigma_T$

3.2 SoS Agent Behavior

SoS agent is responsible for the overall SoS engineering activity and coordinates with individual system agents to achieve the desired SoS mission. In the model, it is assumed that an initial SoS mission is already determined and an initial baseline SoS architecture is available. The SoS agent follows the six core SoS engineering activities outlined in the Wave Process Model in order to develop the SoS. The SoS architecture evolves based on the behavior of individual systems as well as changes in the external environment.

Initiate SoS – During the initialization phase, the wave interval, the time interval from one wave to next, is determined. At each wave interval time, the SoS agent identifies SoS target measures which comprises desired SoS capabilities and SoS performance parameters for these capabilities in order to meet mission objectives. Since some of the capabilities may have higher priority levels than others, weighted value of each capability is also identified at this phase. Table 2 summarizes the abstracted model elements in mathematical notation.

Table 2: Initiate SoS

Simulation time: t
Wave interval: epic
Wave time: T=epic. t

At Wave time: T=0

Determine SoS desired capabilities:

$$SoS.C_i = (C_1, C_2, ..., C_n)$$

Determine weighted value for each SoS capability:

$$SoS.w_i = (w_1, w_2, ..., w_n)$$

Determine SoS desired performance parameters:

$$SoS.P_i = (P_1, P_2, ..., P_n)$$

Identify initial SoS Target Measures:

$$SoS.M_0 = [a_{ij}]_{n\times 3} \text{ where}$$

$$a_{i1} = SoS.C_i,\ a_{i2} = SoS.P_i, a_{i3} = SoS.w_i$$

Conduct SoS Feasibility Analysis –The SoS agent allocates SoS capabilities to individual systems or group of systems. This allocation identifies interface and functionality requirements for individual systems. This allocation defines a baseline SoS architecture. The program management measures such as schedule, funding are also identified. The SoS baseline architecture and program measures information is sent to individual systems as a connectivity request to the SoS architecture. Individual systems should evaluate whether they can develop the requested interface with other systems and capabilities in the given deadline and funding. Table 3 summarizes the abstracted model elements in mathematical notation.

Table 3: Conduct SoS Analysis

Identify set of individual systems to satisfy the target SoS measures:

$$SoS.M_0 \rightarrow System.S_i = (S_1, S_2..., S_n)$$

Allocate SoS capabilities to individual systems:

For i=1..n
  For j=1..n

$$SoS.Callocated_i = (C_i, S_i, S_j) \text{ where}$$

$$S_i \rightarrow S_j \text{ and } S_i \neq S_j$$

Determine deadline for each allocated SoS capability:

$$SoS.d_i = (d_1, d_2..., d_3)$$

Determine funding for each allocated SoS capability:

$$SoS.f_i = (f_1, f_2..., f_3)$$

Define initial baseline SoS Architecture:

$$SoS.A_0 = [a_{ij}]_{n\times n} \text{ where } a_{ij} = (SoS.Callocated_i)$$

Send SoS Connectivity Request to individual systems:

$$SoS.R_i = f(SoS.A_0, SoS.f_i, SoS.d_i)$$

Develop and Evolve SoS Architecture – The SoS agent updates the baseline SoS architecture based on information received from individual systems. Individual systems may decide to cooperate at the requested deadline, may decide to cooperate at a later time or may decide to not cooperate at all depending on their motivation. These decisions will affect the SoS architecture evolution. At this step, based on information received from individual systems, the expected SoS architecture at the end of the wave cycle is updated. Table 4 summarizes the abstracted model elements in mathematical notation.

Table 4: Develop and Evolve SoS Architecture

Receive information from individual systems (see Section 3.3, Table 8):

$$System.Information_i$$

Architecture update factor:

$$Beta_T = f(System.Information_i)$$

Expected SoS architecture at wave time T:

$$SoS.A_T = SoS.A_0 + Beta_T$$

Plan SoS Update – At the end of the wave cycle, the SoS agent evaluates changes in the external environment. The SoS target measures and wave interval for the next cycle is updated based on environment changes and architecture gaps analysis. The gap analysis is also conducted at the end of the wave cycle during the SoS implementation step which is described in the following step. Table 5 summarizes the model elements in mathematical notation.

Table 5: Plan SoS update

At wave time T:

- Adjust/update SoS Target Measures:

Capability update factor

$$SoS.\Delta C_i = (\Delta C_1, \Delta C_2, ..., \Delta C_n)$$
$$SoS.\Delta C_i = f(E_t, SoS.Gap_T)$$

Performance update factor

$$SoS.\Delta P_i = (\Delta P_1, \Delta P_2, ..., \Delta P_n)$$
$$SoS.\Delta P_i = f(E_t, SoS.Gap_T)$$

SoS Target measures update factor

$$SoS.Alpha_T = [a_{ij}]_{n\times 2} \text{ where}$$
$$a_{i1} = SoS.\Delta C_i \text{ and } a_{i2} = SoS.\Delta P_i$$

at T=0

$$SoS.Alpha_T = 0$$

SoS Target measures at time T:

$$SoS.M_T = SoS.M_0 + SoS.Alpha_T$$

- Adjust wave rhythm interval

$$Epic = f(E_T, SoS.Gap_T)$$

- Adjust budget/schedule for allocated capabilities

$$SoS.d_i = f(E_T, SoS.Gap_T)$$
$$SoS.f_i = f(E_T, SoS.Gap_T)$$

Implement SoS – At the end of the wave cycle, the current SoS architecture is evaluated against initial SoS baseline architecture to identify the functionality and performance gaps. This step is an input to planning SoS update step. Table 6 summarizes model elements in mathematical notation.

Table 6: Implement SoS architecture

At wave time T:

Gap analysis:

$$SoS.Gap_T = f(SoS.M_T - SoS.M_{T-1})$$

Continue SoS analysis – The next wave cycle of the SoS development starts once the SoS target measures and wave interval time are updated.

3.3 Individual System Behavior

Individual systems receive request for connectivity to SoS architecture. Since each system is independent and has its own goals and motivations, the system has the option to cooperate or not cooperate with the SoS agent. The decision depends on several factors including system's willingness to cooperate which measures the degree of selfishness of the individual system to be part of the SoS, and system's ability to cooperate which depends on system's resources that will allow the system to be part of the SoS. If individual system decides to cooperate, it sends information to the SoS agent on the probability of meeting the requested capability at the given deadline. If individual system decides to not cooperate, it has the option of requesting a later deadline to provide the capability. Table 7 and Table 8 summarize the abstracted model elements in mathematical notation for individual systems.

Table 7: Evaluate SoS Connectivity Request

Individual system: $System.S_i$

System performance: $System.p_i$

System capability: $System.c_i$

Willingness to cooperate: $System.willingness_i$

Ability to cooperate: $System.ability_i$

-Receive Connectivity Request from SoS agent:

$SoS.R_i$

-Evaluate SoS request:

$$System.coop_i = f(System.willingness_i, System.ability_i, SoS.R_i)$$

$$System.coop_i = \begin{cases} 1 \text{ if cooperate} \\ 0 \text{ if not cooperate} \end{cases}$$

Table 8: Reply back to SoS agent

If $System.coop_i = 1$

$$System.Information_i = (System.c_i, System.p_i, System.av_i)$$

where

$$System.av_i = P(SoS.R_i)$$

else

Time to cooperate:

$$System.cooptime_i = t \text{ where } t > SoS.d_i$$

4 Conclusions

This paper outlined a generic agent-based model for analyzing behavioral dynamics of SoS acquisition based on Wave Process Model. Modeling key aspects of such an intricate process is challenging but necessary for understanding the behavioral dynamics. Future research will focus on generation of models for various SoS acquisition applications based on this generic model. It is envisioned that these models will provide insights into strategies for improving the SoS engineering process.

5 Acknowledgment

This material is based upon work supported, in whole or in part, by the U.S. Department of Defense through the Systems Engineering Research Center (SERC) under Contract H98230-08-D-0171. SERC is a federally funded University Affiliated Research Center managed by Stevens Institute of Technology.
Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

References

[1] J. Dahmann, G. Rebovich, R. Lowry, J. Lane and K. Baldwin, "An implementers' view of systems engineering for systems of systems" Systems Conference (SysCon), 2011 IEEE International, pp. 212-217, April 2011.

[2] J. Dahmann, J. Lane, G. Rebovich, K. Baldwin, "A model of systems engineering in a system of systems context" Proceedings of the Conference on Systems Engineering Research, LA, CA, April 2008.

[3] J. Lane, and J. Dahmann, "Process evolution to support system of systems engineering" ULSSIS'08, Leipzip, Germany, May 2008.

[4] R. Creel, and B. Ellison, "System-of-systems influences on acquisition strategy development" https://buildsecurityin.us-cert.gov/bsi/articles/best-practices/acquisition/981-BSI.html, 2008, last accessed March 2012.

[5] J. P. Dauby and S. Upholzer, "Exploring Behavioral Dynamics in Systems of Systems", in Complex Adaptive Systems, Editor C. Dagli, Procedia Computer Science Volume 6, Page 34-39, Elsevier 2011.

[6] E. Bonabeau "Agent based Modeling: Methods and Techniques for Simulating Human Systems," Proceedings of the National Academy of Sciences, Vol. 99, pp. 7280-7287, 2002.

[7] Systems Engineering Guide for System of Systems, Version 1.0, 2008, http://www.acq.osd.mil/se/docs/SE-Guide-for-SoS.pdf, last accessed May 2012.

[8] N. Kilicay-Ergin, D. Enke, and C. Dagli, "Models of Trader Behavior and Analysis of Financial Market Dynamics" International Journal of Knowledge-based and Intelligent Engineering Systems, Vol. 16, pp. 99-116, 2012.

[9] D. D. Dudenhoeffer, and M. P. Jones, "A Formation Behavior for Large-scale Micro-robot Force Deployment" Proceedings of the Winter Simulation Conference, Orlando, FL, 2000.

[10] W. E. Weiss, "Dynamic security: An agent-based model for airport defense" Proceedings of the Winter Simulation Conference, Miami, FL, pp. 1320-1325, 2008.