Journal of Systems Engineering and Electronics
Vol. 34, No. 3, June 2023, pp.627 – 640

A goal-based approach for modeling and simulation of different types of system-of-systems

FENG Yimin[1], ZHOU Chenghu[2], ZOU Qiang[1,*], LIU Yusheng[1],
LYU Jiyuan[3], and WU Xinfeng[3]

1. State Key Lab of CAD&CG, Zhejiang University, Hangzhou 310027, China;
2. Xi’an Aerospace Propulsion Institute, Xi’an 710100, China;
3. China Astronaut Research and Trainning Center, Beijing 100094, China

Abstract: A system of systems (SoS) composes a set of independent constituent systems (CSs), where the degree of authority to control the independence of CSs varies, depending on different SoS types. Key researchers describe four SoS types with descending levels of central authority: directed, acknowledged, collaborative and virtual. Although the definitions have been recognized in SoS engineering, what is challenging is the difficulty of translating these definitions into models and simulation environments. Thus, we provide a goal-based method including a mathematical baseline to translate these definitions into more effective agent-based modeling and simulations. First, we construct the theoretical models of CS and SoS. Based on the theoretical models, we analyze the degree of authority influenced by SoS characteristics. Next, we propose a definition of SoS types by quantitatively explaining the degree of authority. Finally, we recognize the differences between acknowledged SoS and collaborative SoS using a migrating waterfowl flock by an agent-based model (ABM) simulation. This paper contributes to the SoS body of knowledge by increasing our understanding of the degree of authority in an SoS, so we may identify suitable SoS types to achieve SoS goals by modeling and simulation.

Keywords: simulation, systems agent-based modeling, systems-of-systems (SoS), systems thinking.

DOI: 10.23919/JSEE.2023.000084

1. Introduction

A system of systems (SoS) comprises a collection of independent systems that interoperate together to achieve overarching capabilities [1,2]. Applications of SoS can be found in diverse domains, such as military [1], aerospace [3], health care [4], transportation [5], and energy [6], among others. Compared to the traditional system in the field of systems engineering (SE), the development of an SoS in the field of SoS engineering (SoSE) is challenged by increasing complexity results from the independence of constituent systems (CSs) [7,8]. The challenge prompts the need for two aspects of efforts: analyzing the independence of CSs and modeling methods for SoS. On the one hand, as each CS has managerial and operational independence [9,10], an SoS can be designed by considering four types (archetype, category, classification, typology) based on the degree of authority (managerial control) [11]. Understanding SoS types is necessary to provide a framework for understanding SoS [12−14] and supports the development of SoS [15−17]. On the other hand, model-based systems engineering (MBSE), which focuses the SE process on well-constructed models that capture the essence of the system, can help address the complexity of SoS [18,19]. There has been considerable interest in using model-based techniques for SoSE [20,21]. For effective model-based SoS engineering (MBSoSE), Nielsen et al. [8] identified different dimensions, one of which is the category of SoS, for positioning an MBSoSE approach. They point out the need for SoS types to guide the development of an SoS because the degree of managerial control determines the adaptability and collaboration of each CS in terms of requirements, interfaces, data formats, and technologies [8]. In a word, the need for understanding and modeling SoS types to enrich SoS theory and SoS practice is apparent. However, the problem is how to provide a well-defined definition that can be translated into more effective models and simulations of SoSs.

A basic understanding of the SoS types can be obtained directly through the most popular definition of SoS types

---

Manuscript received March 15, 2022.

*Corresponding author.

This work was supported by the National Key Research and Development Program of China (61873236), the Natural Science Foundation of Zhejiang Province (LZ21F020003; LY18F030001), and the Civil Aerospace Pre-research Project (D020101).

[10,11]. The definition provides a text-based interpretation of the degree of SoS authority. However, because the definition lacks a quantitative judgment basis, it can be challenging to determine the SoS type when facing a practical example of an SoS [22]. Therefore, a rigorous theoretical foundation with unequivocal concepts is needed to avoid vagueness and subjectivity [22].

Another kind of understanding of the SoS types can be obtained indirectly through the well-defined theory of SoS characteristics because the SoS type is the further standardization based on SoS characteristics [22]. There is a broad agreement on SoS characteristics that can help recognize or realize an SoS [23]. Subsequently, one can then propose a mathematical framework to translate some concepts of SoS characteristics into an unambiguous theoretical model based on set theory, aiming to rigorously define the SoS characteristics [24−26]. However, the existing definitions of SoS types usually rely heavily on the text-based description without taking full advantage of the theoretical models of SoS characteristics. What is missing is a further explanation of the relationship between SoS characteristics and SoS types; this connotes that more significant effort may be required to explore the close relations between SoS characteristics and SoS types.

As mentioned previously, the existing definition of SoS types is based on the degree of authority [10,11], and SoS development requires the guidance of SoS types [8]. SoS development prioritizes achieving SoS goals and then satisfies CS goals as much as possible [27]. Thus, we argue that the ultimate purpose of SoS authority is to achieve SoS goals, and there is a relationship between the degree of authority and both SoS goals and CS goals. Using goals brings several advantages compared to using characteristics. First, SoS goals are generally explicit and convenient to detect except in a virtual SoS where SoS goals are implicit [13]. In particular, when there is no explicit central management authority, the degree of authority is sometimes implicit and difficult to detect, such as in the example of swarm robots [28]. Second, goals can be measured quantitatively and objectively by the well-defined goal statement [29,30]. Next, goals are essential elements in SoS modeling, such as requirement modeling [31] and mission-based approaches [32,33], which means that goals can serve as a bridge between SoS theory and SoS practice. Therefore, a goal-based analysis appears to further facilitate the understanding of SoS types and strengthen the link between SoS characteristics and SoS types. Regarding SoS characteristics, the belonging index comprising two parameters, CS goals and SoS goals, indicates the cohesion of an SoS [30]. The challenge is that SoS characteristics can not be used directly to explain the degree of authority. These two parameters still leave room for interpreting how much a CS will comply.

In this paper, a goal-based approach is proposed to model and simulate different types of SoS. First, the theoretical models of both CS and SoS are constructed by introducing a new concept called potential. Second, goal states influenced by SoS characteristics are described. Then, difference among four SoS types based on the transitions among the combinations of goal states is unambiguously demonstrated. According to the analysis of SoS types, the difference between acknowledged SoS and collaborative SoS is recognized based on the agent-based model (ABM). The goal-based approach is valuable in providing a contribution to SoS theory-related aspects by increasing our understanding of SoS type and providing a mathematical baseline to translate the theoretical definition into a more effective ABM simulation of SoS, which helps identify the suitable SoS types to achieve SoS goals.

We organize the remainder of this paper as follows. Section 2 reviews the related work. Section 3 introduces concepts of various goals and constructs the CS and the SoS theoretical models based on set theory. Section 4 analyzes the influence of SoS characteristics on goals by some possible combinations of goal states. Section 5 explains the SoS type through the transitions of the combinations. Section 6 implements the proposed approach using ABM to simulate the migrating waterfowl. Section 7 discusses the results. Finally, Section 8 summarizes the conclusions.

2. Background and literature review

Related work to this study can be classified into three topics, which are the explanations of SoS types, SoS characteristics and theoretical models, as well as modeling of SoS types.

2.1 Explanations of SoS types

Maier [9] described three different SoS types “directed, collaborative, and virtual.” Dahmann and Baldwin [11] added an intermediate type called acknowledged between directed and collaborative. The degree of SoS’s authoritative control over CSs decreases from directed SoS to virtual SoS. These four SoS types may have implications for the relevant engineering processes because the authority relationships between SoSs and CSs affect how SoS engineering (SoSE) can be implemented [2]. To illustrate, a directed SoS that has the designated SoS manager with complete control over CSs needs to consider the technical features of the CSs to achieve SoS goals. Meanwhile, an acknowledged SoS or a collaborative SoS that does not directly authorize CSs needs to consider how the CSs can participate in SoS decisions that impact CS goals [2].

The primary issue is that the aforementioned text-based interpretation of the degree of authority has vagueness and subjectivity. Specifically, due to the lack of rigorous models [8] to describe the degree of authority, determining SoS types usually relies heavily on human experience, which may result in misclassification [10] or even no classification [12]. For example, suppose a collaborative SoS is misclassified as directed. In that case, the SoS operators will have less control over the purpose than they think, resulting in collective operations across administrative boundaries that will not reliably occur in practice [10]. As a result, perhaps only experienced engineers can rely on their own experience to understand and determine the SoS type. Thus, there have been various attempts to explain these types better, such as using bi-directional arrows between the SoSE team and CSs [34] and multi-dimensional classification [16]. However, ambiguity and subjectivity still exist due to the qualitative description without any rigorous theoretical model.

2.2 SoS characteristics and theoretical models

Nielsen et al. [8] discussed Maier's characteristics (acronym "OMGEE") [9] and Boardman and Sauser's characteristics (acronym "ABCDE") [23,35]. Maier's characteristics deal with the lack of a shared agreement on the SoS definition. Boardman and Sauser's characteristics distinguish SoSs in SoSE from conventional systems in SE, using the following five aspects [25,36]. Autonomy (A) is the ability of a CS to achieve its own CS goals with limits without the control of the SoS or other CSs. Belonging (B) is the ability of a CS to receive the benefit of its own CS goals and contribute to SoS goals or goals of other CSs. Connectivity (C) is the capability to form connections as needed to benefit each other. Diversity (D) emphasizes the difference in CS goals among various CSs. Emergence (E) means the cumulative actions and interactions between the constituents of an SoS give rise to behaviors attributed to the SoS as a whole. Based on the definitions above, SoS types appear to correlate most with authority and belonging due to the correlation between CS goals and SoS goals.

Felder and Baldwin [29] calculated the belonging index (gains/contributions) by a mathematical expression. However, there is no further segmentation in these two goal dimensions. As a result, it is difficult to support the classification of multiple SoS types. For this reason, expanding these two parameters is necessary to analyze quantitatively. We construct the SoS theoretical model by introducing a new concept called "potential" for further segmentation of CS goals and SoS goals. We propose the term "goal state" to represent scalars in two dimensions of goals, which indicates that a goal is achieved by what kinds of function and potential. As the scales of the two goals dimensions increase, the combinations of different states are likely to be sufficient to distinguish SoS types.

2.3 Modeling of different SoS types

Baldwin and Felder quantified a metric of SoS feature attribution for a collaborative SoS migrating waterfowl flock [29] and a directed SoS air defense scenario [30]. They contended that the essential difference between collaborative SoS and directed SoS is a governance scheme in which a central authority determines individual systems' level and type of participation [30]. However, the distinction between these two SoS types is only directly reflected in the concept introduction but not directly reflected during the modeling and simulation process. The SoS type is only used as a fundamental concept without practical guidance for modeling and simulation due to the insufficiency of qualitative description. Soyez et al. [37] proposed a multilevel agent-based method to model a directed SoS or acknowledged SoS, using the case of intelligent autonomous vehicles (IAVs) and formalizing the five characteristics. Their work mainly deals with a directed or acknowledged SoS, which may be applied to other types of SoSs if further studying the degree of authority. Darabi and Mansouri [28] used multi-attribute utility theory (MAUT) to represent the utility functions of CSs and SoS. The swarm robot used in the case consumes resources to find targets, but the researchers did not specify which type this SoS is, making it very difficult to extend their findings to all four SoS types. The swarm robots were designed to achieve SoS goals first and then CS goals. However, they stated that the robots achieved CS goals and SoS goals without priority in their follow-up discussion. Thus, the relationship between SoS goals and CS goals is ambiguous. Seo et al. [38] analyzed all four SoS types and used a probabilistic variable to model three of them except virtual SoS. However, how to define the value of the probabilistic variable still leaves much room for interpretation.

There are two mainstream ideas for modeling and simulation of SoS. Baldwin et al. [39] compared ABM and event-based modeling (EBM) and suggested that ABM is more suitable for describing four of the five SoS characteristics autonomy, belonging, connectivity, and diversity. For example, different examples of SoS use ABM, such as collaborative SoS birds flock [29], directed SoS air defense scenario [30], directed or acknowledged SoS intelligent autonomous vehicles [37]. ABM deals with the complexity of interactions between CSs and represents an

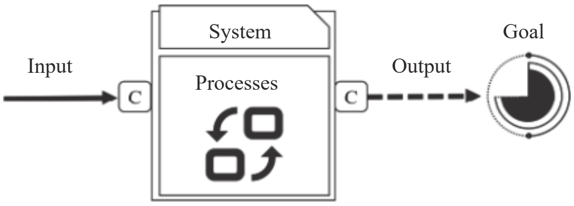
SoS through a few rules [40]. This paper analyzes all possible situations of the interactions between CSs through the goal-based description. Thus, the goal-based description can guide the development of each agent's rules.

3. Theoretical models

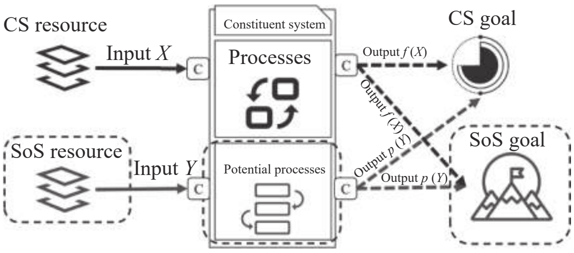
The SoS theoretical model is based on the CS theoretical model. The existing theoretical model [24] can only describe a limited number of goal states of CS and SoS, which is not sufficient to specify the SoS types. We construct the CS theoretical model, on which the SoS theoretical model is constructed, to support the subsequent description of the goal state.

3.1 CS theoretical model

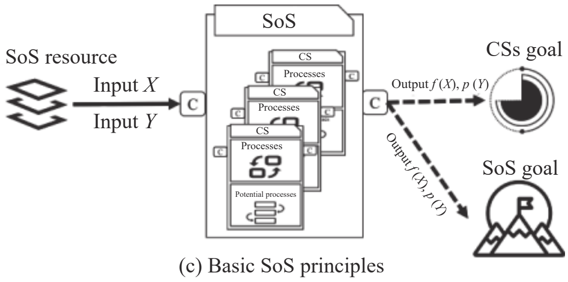
We unify the terminology used in the paper as "CS" for a system in SoSE and "system" for a system in SE. Due to the difference between SoS and systems, the traditional system principles [41] still have some room for interpretation when describing CS in SoS. Baldwin and Sauser [26] described the CS theoretical model and SoS characteristics based on set theory and first-order predicate logic. However, it is not sufficient to explain and describe SoS types directly. Axelsson [22] proposed a set of terms to describe the different states of CSs in an SoS. For example, "Passive CS" means the CS has participated in an SoS, and "Prepared system" means the CS has not participated in an SoS. These terms can help understand the difference between a CS and a system and better develop theoretical models. This paper introduces a new concept to the most basic form of the traditional system called "potential" for describing CS. More specifically, based on the original function of CS as a system in SE, we propose the potential as an additional type of the functions of CSs in SoSE. The potential is an inherent attribute of the CS, whether the CS has participated in an SoS or not. Notably, the potential can only be activated by another CS's resources or outputs instead of the CS's resources in the environment, which is different from the original function of CSs. We use "latent potential" to term the inactivated potential and "realized potential" to term the activated potential. Accordingly, we argue that a CS's functions include the original functions in SE and the potential in SoSE. In the most basic form [41], a system, S_i, takes input and transforms it via a set of processes into output to attain goals set, G_i, as denoted in Fig. 1(a). Now we can describe the basic CS principles as a CS S_i transforms inputs (or resources) X_i into outputs through a series of processes (functions) $f_i(x)$ to achieve CS goals and SoS goals G_i. Additionally, the CS transforms other inputs Y_i into outputs through another set of processes (potential) $p_i(x)$ to achieve CS goals and SoS goals.



(a) Basic system principles in SE



(b) Basic CS principles in SoSE



(c) Basic SoS principles

Fig. 1 Basic principles of the system, CS and SoS

Thus, we can describe the CS theoretical model based on set theory as follows:

$$S_i = \{X_i, F_i, P_i, G_i, C_i\}, \tag{1}$$

$$\begin{cases} F_i' \to G_i^+, \ F_i' \subseteq F_i \\ P_i' \to G_i^-, \ P_i' \subseteq P_i \\ G_i^+ \cup G_i^- = G_i \end{cases}. \tag{2}$$

As shown in Fig. 1(b), a set of CS functions is represented as F_i, which forms the CS's process and transforms inputs X_i into outputs that achieve the set of CS goals G_i. Similarly, after participating in an SoS, the CS has a specific role, and the new set of functions for achieving CS goals and SoS goals is denoted as potential P_i. Another process formed by the potential transforms inputs Y_i received from the SoS after participating in an SoS into outputs. These outputs attempt to achieve SoS goals and may also contribute to achieving CS goals set G_i. And G_i is composed of the goals set G_i^+ and the goals set G_i^-. Notably, G_i^+ can be achieved by the original functions, and G_i^- can be achieved by the realized potential after the CS joins an SoS, as shown in (2). Moreover, the set of connections (or interfaces) of a CS C_i allows outputs from one CS to become inputs to another CS.

According to the above definition of the potential, the SoS resources (inputs), potential processes, and the SoS goals (outputs) achieved by the potential after CSs join an SoS are identified by the dashed boxes in Fig. 1(b). In other words, CSs cannot activate potential processes independently without participating in SoS.

3.2 SoS theoretical model

An SoS is a composite system that contains multiple CSs and acts as a whole with SoS goals. Based on the CS theoretical model, we describe the SoS theoretical model as illustrated in Fig. 1(c), which analyzes and explains the CS goals and SoS goals in detail.

$$S^* = \{S_1, S_2, \cdots, S_n, G^*\}, \tag{3}$$

$$F_i = \begin{cases} F_i' \rightarrow G_i^+ \\ F_i^* \rightarrow \text{part of } G^* \end{cases}, \tag{4}$$

$$P_i = \begin{cases} P_i' \rightarrow G_i^- \\ P_i^* \rightarrow \text{part of } G^* \end{cases}, \tag{5}$$

where G_i^* are the SoS goals achieved by all CSs in the SoS. part of G_i^* indicates the contribution of individual CSs to the SoS goals, where individual CSs can only achieve a portion of the SoS objectives; G_i^+ are the CS goals that the CS can achieve by itself alone; G_i^- are the CS goals that cannot be achieved by the CS and can be achieved when participating in the SoS; F_i are the functions that the CS can operate independently by itself only; F_i' are the functions that can be run independently and are only used to implement G_i^+; F_i^* are the functions that can be run independently of the CS, which are used only for SoS goals; P_i are the functions that the CS cannot run by itself alone, which are activated after participating in SoS; P_i' is the potential activated after CS participates in the SoS, which is used only for the CS goals; P_i^* is the potential activated after CS participates in the SoS, which is used only for the SoS goals.

Considering that we have introduced the relationship between CS goals and potential in (2), we propose that the SoS goals set G_i^* can be achieved by CS functions set F_i^* and CS potential set P_i^* in (4) and (5). We regard functions as F_i' for achieving CS goals and F_i^* for achieving SoS goals. Likewise, we regard potential as P_i' and P_i^*. Potential can only be activated to achieve SoS goals after participating in an SoS, whereas functions F_i' and F_i^* can produce outputs whether the CS has participated in an SoS or not. Notably, the outputs produced by F_i' of a CS benefit the CS itself G_i^+. Meanwhile, the outputs produced by F_i^* of a CS can benefit G_i^- of other CSs in the SoS only after the CS participates in an SoS, which is equivalent to benefiting the SoS goals, as shown in (4). As shown in (2), the benefit for G_i^- by P_i' is the motivation for a CS to participate in an SoS. Similarly, the benefit for SoS goals by F_i^* and P_i^* is the motivation for an SoS to accept the participation of a CS, as shown in (4) and (5). In other words, discovering and exploiting the potential and F_i^* of a CS may be feasible to realize the SoS capabilities.

4. Analysis of goal states

According to the above theoretical models, there are two dimensions of goals, SoS goals and CS goals, and four kinds of functions, F_i', F_i^*, P_i', and P_i^*, can be used to achieve goals. Since a goal can be achieved through different cases of functions, which are illustrated by using an example of swarm robots [28], we regard these cases as scalars representing different states in each dimension of goals. Then we combine the scalars in the two dimensions to get different combinations of states. Since autonomy and belonging can influence the achievement of CS goals and SoS goals, we analyze all possible combinations of goal states and use a migrating waterfowl flock to explain the combination, which supports the subsequent analysis of the transitions between these combinations in the next section.

4.1 Influence of SoS characteristics on goal state

In the process of SoS evolution, individual CSs are committed to achieving CS goals and affect SoS goals under the influence of SoS characteristics. Since the relationship between autonomy and belonging influences SoS types, it is necessary to analyze all the possible effects of SoS characteristics on the goals. Thus, an individual CS must consider all possible realizations of CS goals and SoS goals when the specific scenario is not considered. To support the description of these cases, we use an example of swarm robots, which is an instance of an SoS [28,42] and is easy to illustrate the discussed cases in this context.

There are two possible cases for an individual CS to achieve CS goals:

(i) The functions are used for CS goals, and the potential is activated to be used for CS goals.

$$\text{Case A:} F_i' \rightarrow G_i^+, P_i' \rightarrow G_i^-$$

(ii) Only the functions are used for CS goals.

$$\text{Case B:} F_i' \rightarrow G_i^+$$

In the SoS of swarm robots, the robots can share their resources with other robots, and they search for resources to survive and search for targets to achieve the SoS goal

[28]. Thus, F_i' is that a robot attains resources by finding and moving, G_i^+ is to attain resources for surviving by itself. Moreover, P_i' is that a robot receives resources by sharing with other robots who have more resources, and G_i^- is to get resources from others. Case A means that the robot gets resources by finding targets and sharing them with others. Case B means that the robot only attains resources by itself for some reason. For example, the robot shares its resources with others with fewer resources or does not encounter other robots during moving.

There are four possible cases for an individual CS to achieve SoS goals:

(i) Both the functions and the potential are used for SoS goals.

Case 1: $F_i^* \rightarrow$ part of G^*, $P_i^* \rightarrow$ part of G^*

(ii) Only the potential is used for SoS goals.

Case 2: $P_i^* \rightarrow$ part of G^*

(iii) Only the functions are used for SoS goals.

Case 3: $F_i^* \rightarrow$ part of G^*

(iv) Nothing is used for SoS goals.

Case 4: $0 \rightarrow$ part of G^*

In the SoS of swarm robots, F_i^* is that a robot searches for targets, the SoS goal G^* is to find as many targets as possible. P_i^* is that a robot loses resources by sharing with other robots who have fewer resources so that other robots can use more resources to search for targets. Case 1 means that a robot searches for targets by itself and gives its resources to other robots. Case 2 means that a robot only gives its resources to other robots and does not search for targets by itself for some reasons, for example, the robot may have trouble moving. Case 3 means that a robot only searches for targets by itself and does not give others resources for some reasons, for example, the robot does not encounter other robots during moving. Case 4 means that the robot does not contribute to the SoS goal, such as the robot only searching for resources to survive rather than targets.

Thus, combining the preceding two kinds of cases, now we can get $2 \times 4 = 8$ combinations:

(i) Combination A1: The potential is activated to be used for both CS goals and SoS goals.

$$\mathrm{A1} = \begin{cases} F_i' \rightarrow G_i^+, P_i' \rightarrow G_i^- \\ F_i^* \rightarrow \text{part of } G^*, P_i^* \rightarrow \text{part of } G^* \end{cases}$$

(ii) Combination A2: The potential is activated to be used for both CS goals and SoS goals. Meanwhile, functions are only used for SoS goals.

$$\mathrm{A2} = \begin{cases} F_i' \rightarrow G_i^+, P_i' \rightarrow G_i^- \\ P_i^* \rightarrow \text{part of } G^* \end{cases}$$

(iii) Combination A3: The potential is activated to be used for CS goals but not for SoS goals. Meanwhile, functions are used for SoS goals.

$$\mathrm{A3} = \begin{cases} F_i' \rightarrow G_i^+, P_i' \rightarrow G_i^- \\ F_i^* \rightarrow \text{part of } G^* \end{cases}$$

(iv) Combination A4: Potential is activated, is used for the CS goal and the SoS goal, but nothing is used for the SoS goals.

$$\mathrm{A4} = \begin{cases} F_i' \rightarrow G_i^+, P_i' \rightarrow G_i^- \\ 0 \rightarrow \text{part of } G^* \end{cases}$$

(v) Combination B1: The potential is activated, not for the CS goal but for the SoS goal. The functions are used for the SoS goal.

$$\mathrm{B1} = \begin{cases} F_i' \rightarrow G_i^+ \\ F_i^* \rightarrow \text{part of } G^*, P_i^* \rightarrow \text{part of } G^* \end{cases}$$

(vi) Combination B2: The potential is activated and is not used for the CS goal but for the SoS goal.

$$\mathrm{B2} = \begin{cases} F_i' \rightarrow G_i^+ \\ P_i^* \rightarrow \text{part of } G^* \end{cases}$$

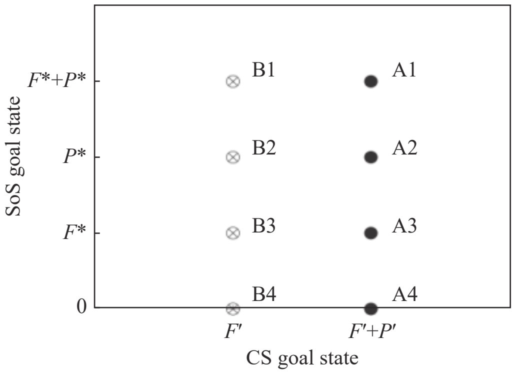
(vii) Combination B3: The potential is not activated. Functions can be used for SoS goals.

$$\mathrm{B3} = \begin{cases} F_i' \rightarrow G_i^+ \\ F_i^* \rightarrow \text{part of } G^* \end{cases}$$

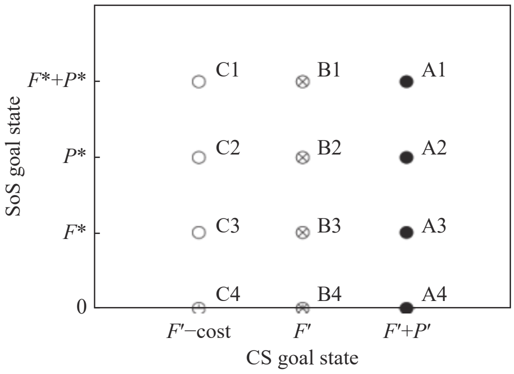
(viii) Combination B4: Potential cannot be activated when the CS is not added to or removed from the SoS. Function produces F_i' and F_i^*, but P_i' cannot be utilized.

$$\mathrm{B4} = \begin{cases} F_i' \rightarrow G_i^+ \\ 0 \rightarrow \text{part of } G^* \end{cases}$$

The preceding eight combinations can be represented by a two-dimensional coordinate system, as shown in Fig. 2(a). Notably, the horizontal coordinate represents the benefit to CS goals, and the vertical coordinate represents the benefit of the CS to SoS goals. Moreover, both dimensions are scaled according to the increase in goal benefit. The difference between the two dimensions is that the CS goal state generally starts from F_i' because CS functions are actual, while the SoS goal state starts from 0 because the SoS is imaginary. The effect of a single CS on the SoS goal is 0 when the CS is not added to the SoS. The relative measurement of F_i^* and P_i^* in the SoS goal should be determined according to the specific example. In this paper, F_i^* is smaller than P_i^* by default, which can be considered that F_i^* is easily accessible, and P_i^* is not easily accessible. Furthermore, P_i^* and P_i' may not have fixed values.



(a) Basic eight combinations of goal states



(b) Expanded twelve combinations of goal states

○ : Pay cost; × : Only function; ● : Active potential.

Fig. 2 Combinations of goal states

The above is simply considering all possible combinations of goal states in collaborative SoS and acknowledged SoS, where CS goals will not benefit less after joining the SoS. In addition, when considering the influence of authority in a directed SoS, we can analyze the respective bottom lines of CS and SoS.

(i) CS bottom line: Autonomy without the influence of authority, which is a bottom-up process, such as no obstruction to $F_i' \rightarrow G_i^+$. On the contrary, compulsory belonging under the influence of authority is a top-down process. Only in this case can $F_i' \rightarrow G_i^+$ be obstructed.

(ii) SoS bottom line: No obstruction to G^*. When a new CS participates in the SoS, it cannot obstruct the benefit of SoS goals.

Considering the influence of the authority on CS goals, there will be a new case:

$$\text{Case C}: F_i' - \text{cost} \rightarrow G_i^+.$$

To restrain CS functions, we use subtraction in Case C, and the value of the cost is positive by default, so the range of the cost is $[0, F_i']$. Combining Case C with Case 1 to Case 4 will generate another four combinations C1 to C4, so that altogether there will be $3 \times 4 = 12$ combinations, as shown in Fig. 2(b).

Under the authority of a directed SoS, the autonomy of CS may be limited, so the state of achieving CS goals after joining the SoS could be lower than that when leaving the SoS, and the benefit of SoS goals must increase. For example, Baldwin and Felder [30] use an air defense scenario to illustrate the use of governance in a directed SoS, in which CSs sacrifice performance against CS goals in favor of the SoS goal.

4.2 Illustration of static aspects

For better comparison, we call the combinations of cases the static aspect of the proposed approach and the transitions between the combinations as the dynamic aspect of the approach. This work uses a migrating waterfowl flock to explain the static aspects of goal state combinations. In the bird flock, what benefits SoS goals is the aerodynamic drag reduction advantage produced by birds in the flight formation [29]. It is worth mentioning that participation in the formation flight SoS carries no penalty: the benefit of drag reduction from one member to another is free of cost to the leading bird [29]. We explain the bird flock SoS based on the CS and SoS theoretical models:

(i) G_i^*: The entire formation consumes less and flies farther, and each bird in the formation consumes less and flies farther on average;

(ii) G_i^+: Normal fuel consumption when flying alone;

(iii) G_i^-: Minimizing fuel consumption by obtaining the updraft generated by other birds ahead;

(iv) F_i: Flapping the wings;

(v) F_i': Flying alone;

(vi) F_i^*: Whether a bird joins a formation or flies alone, the bird produces updrafts on both sides of its wings behind; after joining the SoS, the updraft can be utilized by other birds behind;

(vii) P_i: Aerodynamic advantage;

(viii) P_i': Attaining aerodynamic advantage by obtaining the updraft;

(ix) P_i^*: It does not exist in this SoS.

Because no bird in this SoS carries a penalty, which indicates no CS pays a cost, the combinations that fit the migrating waterfowl are represented in Fig. 2(a). Each bird can only be one of the following four combinations of goal states at a specific moment;

(i) A3: Middle of the formation;

(ii) A4: End of the formation;

(iii) B3: Leading of the formation;

(iv) B4: Flying alone.

5. Explanation of SoS types

Notably, an SoS is never fully formed or complete [43], which means each bird in the flock SoS may change its position over time. In other words, the combinations of goal states of a specific bird may change among the above four cases. Therefore, we can analyze the direction of change in the goal states during the dynamic transition between each combination to explain the different SoS types.

5.1 Transitions between combinations of goal states

According to the existing definitions of SoS types, the SoS is directed if the CS's goals benefit less after joining an SoS, and the SoS is virtual if there are no explicit SoS goals in an SoS. However, the difference between acknowledged SoS and collaborative SoS is challenging to recognize. In these two SoS types, since all CSs have the chance to benefit from participating in an SoS, these two SoS types may have similar situations of achievement of goals. Therefore, we have introduced the new concept of "potential" to enrich the dimensions of goal and have analyzed various cases and combinations of these cases under the influence of authority on SoS goals and CS goals. Considering the difference between the combinations, the transition from one combination to another will change the goal state in two dimensions. We categorize the changes from two dimensions, then combine the classification results to get multiple cases that specify different SoS types, as depicted in Table 1.

Table 1 SoS classification based on two dimensions

SoS classification		SoS goal dimension	
		$\Delta G^* > 0$	$\Delta G^* = 0$
CS goal dimension	$\Delta G_i > 0$	Collaborative	Collaborative
	$\Delta G_i = 0$	Collaborative	–
	$\Delta G_i < 0$, $\Delta G_i = -P'_i$	Acknowledged	–
	$\Delta G_i < 0$, $\Delta G_i = -\text{cost}$	Directed	–

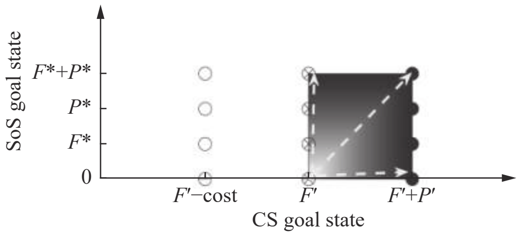
The specific explanation of the method represented in Table 1 is as follows:

(i) Collaborative SoS: If and only if each CS is in one of the following scenarios, the SoS is collaborative. Three scenarios for CS goal states are as follows: the benefit of CS goals increases and the benefit of the SoS goals increases; the benefit of CS goals increases and the benefit of SoS goals remains unchanged; the benefit of CS goals remains unchanged, and the benefit of the SoS goals increases.

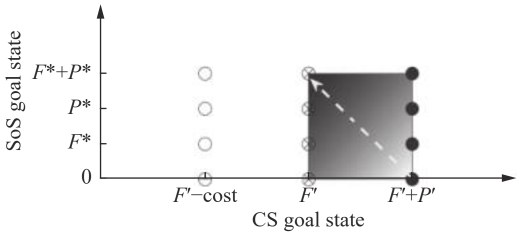
(ii) Acknowledged SoS: If at least one CS is in one of the following scenarios, the SoS is acknowledged. On the one hand, the benefit of at least one CS decreases but does not pay a cost such as $F'_i \rightarrow G^+_i$, and the benefit of SoS goals increases. On the other hand, there is an additional judgment criterion for acknowledged SoS: the three scenarios that should have automatically switched in the collaborative SoS do not switch due to the influence of authority.

(iii) Directed SoS: The benefit of at least one CS decreases, and the CS pays a cost such as $F'_i \rightarrow G^+_i$, but the benefit of SoS goals increases.

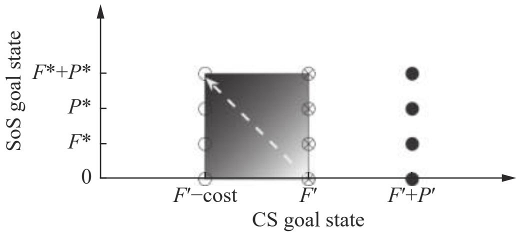
Notably, all CSs in collaborative SoS satisfy the conditions in Table 1. In acknowledged and directed SoS, at least one CS satisfies the conditions. In collaborative SoS, since the SoS will not allow CSs that will decrease the benefit of SoS goals to participate in the SoS, there will be no situation where the benefit of CS goals increases while the benefit of SoS goals decreases. The CS in a virtual SoS only knows about its own goals, not about SoS goals, even not about the goals of other CSs [44]. Therefore, it is impossible to know ΔG^*, that is why Table 1 does not show virtual SoS. We use Fig. 2(b) to plot this approach, as shown in Fig. 3. Under the premise of sufficient resources and conditions, the CS of different SoS will actively move from the lighter color area to the darker colored area.



(a) Collaborative SoS



(b) Acknowledged SoS



(c) Directed SoS

○ : Pay cost; × : Only function; ● : Active potential.

Fig. 3 SoS classification based on transitions

5.2 Illustration of dynamic aspects

To illustrate SoS types based on the dynamic transition between coordinate points on the two-dimensional coordinate system, as shown in Fig. 3, we use the migrating waterfowl SoS with four combinations of goal states as mentioned above.

Collaborative SoS (three scenarios): coordinate points actively switch under sufficient resource conditions.

(i) Vertical up: benefit SoS goals but not CS goals, such as B4→B3 and A4 → A3.

(ii) Top right: benefit SoS goals and CS goals, such as B4 → A3 (generally cannot be achieved directly, as first B4 → A4, and then A4 → A3).

(iii) Horizontal right: benefit CS goals but not SoS goals, such as B4→B4 and B3 → A3 (current SoS merges into another larger SoS).

Acknowledged SoS: coordinate points will switch under authority.

(i) Top left: benefit SoS goals and limit CS goals, but CS does not pay a cost, such as A4 → B3.

(ii) Additional judgment criterion: even when the resource conditions are sufficient, the coordinate points of the three scenarios cannot actively switch as in the collaborative SoS due to the influence of SoS authority.

(iii) Directed SoS: does not exist in this SoS because birds will not pay a cost in any situation.

6. Modeling and simulation

To verify the theoretical models and the static aspect and dynamic aspect of our approach for explaining SoS types, we use a model of migrating waterfowl flock, which is well suited to verify the behavior of CSs and SoS [29].

6.1 ABM modeling approach

We develop modeling and simulation with the ABM method to verify the goal-based definition of SoS types. The ABM description in Table 2 describes the model according to the overview, design concepts, and details (ODD) protocol [45,46], making model descriptions more straightforward and easier to understand. This protocol provides a common format and standard structure to simplify the reading of ABM descriptions.

Table 2 Summary of ODD protocol for ABM

Overview	Design concepts	Details
Purpose: to verify the simulation results when different types of the same SoS are used. **Entities:** fixed number of agents is 30; fixed time step duration of 0.1 s; fixed speed increase/decrease rate is 20%; fixed rotation angle per step is 4°. **State variables:** Facing north/not north; whether the four agent rules are satisfied. **Scales**: 70×70 grids. **Process overview and scheduling:** Collaborative SoS: Each bird starts at a random position and goes out into the world. If the bird cannot see another bird within a limited range of vision, it will continue to fly straight at its normal basic speed. If another bird can be seen, the following four basic rules are followed, given by that order of priority. (i) If it is too far from the nearest visible bird, it will turn toward that bird and approach it at an increased speed. (ii) Once it is close enough to another bird, it randomly moves to one side or the other until its view is no longer obscured. (iii) If it gets too close to another bird, it will slow down. (iv) Once the preceding three conditions are met (the bird has an unobstructed view and is close enough but not too close to another bird), the bird will set its speed and the forward direction of its closest visible neighbor. Acknowledged SoS: Rule 1 and Rule 4 disabled.	**Basic principles:** the model represents a simplified behavior based on natural physical phenomena. **Objectives:** different types of the same SoS. **Stochasticity:** initialized random number of seeds. **Interaction:** agents will acquire traction from other agents in the vicinity. **Adaptation:** agents try to occupy a favorable position and avoid an unfavorable one. **Sensing:** sensing distance of agents is 15-patch; sensing angle of agents is 120° cone range in front of it; obstruction cone of agents is 45°; obstruction distance of agents is 10-patch. **Prediction:** collision point prediction. **Emergence:** flock geometry; SoS goals. **Observation:** the heading of each agent after forming a flight formation.	**Initialization:** the initial position and initial heading are random. **Input data:** no external data. **Submodel 1(solo):** the default state of a bird agent is to be solo. When no other birds are in vision to provide an updraft, a bird cruises towards its destination. **Submodel 2(Alignment as Rule 4):** a bird tends to turn, moving in the same direction as nearby birds. **Submodel 3(Separation as Rule 3):** a bird will turn to avoid another bird that gets too close. **Submodel 4(Cohesion as Rule 1 and Rule 2):** a bird will move towards other nearby birds (unless another bird is too close).

Each agent is born in a random position and moves in a random direction, consistent with the B4 case where a bird is flying alone. Notably, an agent determines the current environment at each time step and thus determines the acceleration and direction of movement. According to the preceding illustration of SoS types, the "potential" of an agent in the bird flock SoS is to obtain an updraft. To realize the "potential," an agent should use its limited vision range to find the location of other agents in the conical region ahead that generate updrafts, then select the closest agent, change its heading, and accelerate to try to approach. According to the scenarios illustrated in Subsection 4.2, we develop four rules ranked in order of priority for CSs in a collaborative SoS, as shown in Table 2. Rule 1 describes the process of “potential” for a bird to obtain the updraft. Rule 2 and Rule 3 aim to adjust the distance among agents in the flock to make the flock look closer to reality. Rule 4 is to maintain the “realized potential.” These rules are consistent with the existing rules for birds flying in formation, which are proposed by Nathan et al. [47] and Felder and Baldwin [29]. In this collaborative SoS, if an agent does not find another agent within its limited field of view, it will keep flying in a straight line at its normal base speed. The above definition of acknowledged SoS indicates that CSs may be barred from benefiting CS goals. In the bird flock SoS, even if an agent finds another agent within its limited field of view, it still keeps flying in a straight line at the normal base speed instead of getting close to another agent to obtain the updraft. To this end, when we simulate the acknowledged SoS, Rule 2 and Rule 3 are enabled while Rule 1 and Rule 4 are disabled.

A simulation stops when the running ticks reach 1 000 000. The simulation environment in this paper has parameters such as the number of agents, the range of limited vision of agents, and the speed of agents. In addition, different settings of parameters will result in different effects on the simulation. Since this paper focuses on different types of SoS, these parameters are set to the fixed values, and the thing that can be changed manually is to control which rule is enabled and disabled.

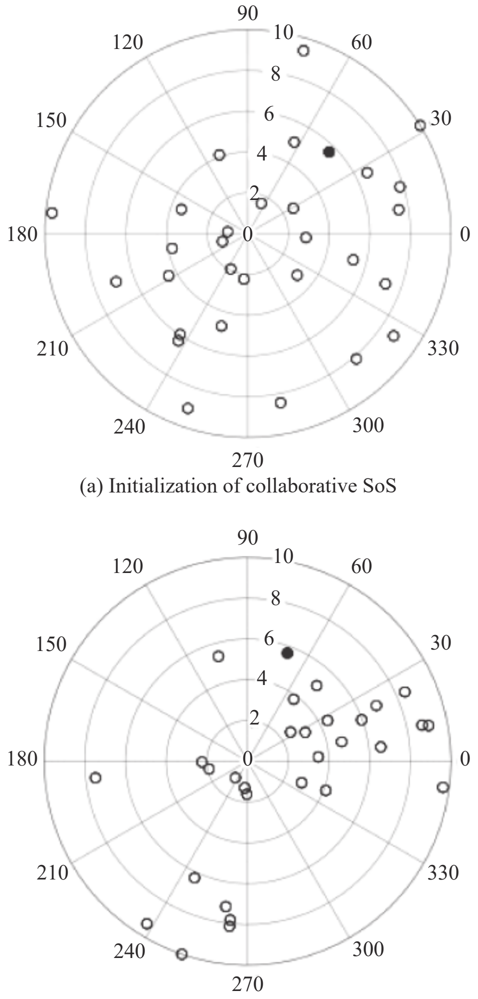
6.2 ABM simulation results

The model is simulated in the agent-based modeling software NetLogo [48]. The simulation output is stored using Microsoft Excel and analyzed by Matlab R2020b.

The model produces realistic formation geometries: One of the goals is to build a model that correctly produces results consistent with the geographic distribution of bird flight formation observed in nature. Our simulation of the bird flock SoS produces two stable flight formation geometries such as the “I” and the “V” as shown in Fig. 4, which are consistent with the reality.

Fig. 4 Vee with birds inside

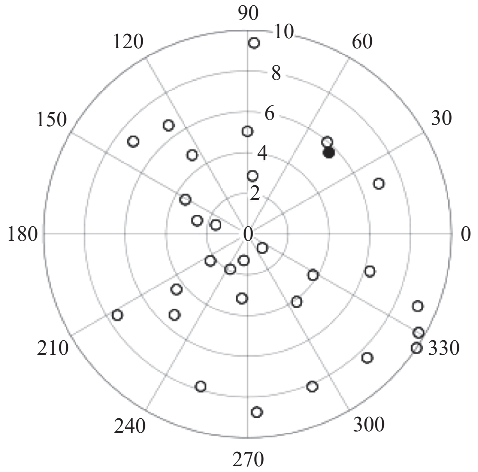
Different types of the flock SoS: Fig. 5 and Fig. 6 compare the collaborative SoS and the acknowledged SoS of the bird flock SoS. The polar coordinates in the figures indicate the heading of each agent. Fig. 5(a) indicates that the heading of each bird is random after initialization. Fig. 5(b) indicates the heading of each bird when the simulation stops. For better observation, the radii of the birds in the polar coordinate are determined by the bird ID generated during initialization, and the radius corresponds to the value in the arithmetic sequence. Fig. 5 and Fig. 6 show that the birds are not facing the same direction after initialization when they do not form a formation. In contrast, when the simulation stops, they form an “I” and “V” formation where the birds in the same formation are facing the same direction. In acknowledged SoS, the solid dots represent the leading birds influenced by authority, and the ID of the leading birds is the median number of all birds’ IDs.



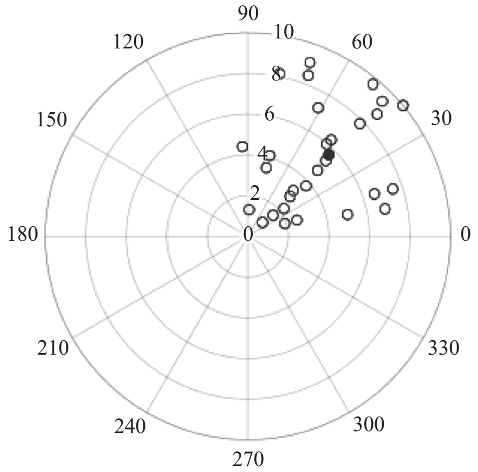
(a) Initialization of collaborative SoS

(b) Ending of simulation of collaborative SoS

Fig. 5 Simulation of collaborative SoS



(a) Initialization of acknowledged SoS



(b) Ending of simulation of acknowledged SoS

Fig. 6 Simulation of acknowledged SoS

7. Discussion

In the collaborative SoS and the acknowledgment SoS, the birds' headings converged more when the simulation stopped than when the initialization was completed, as shown in Fig. 5 and Fig. 6. Besides, the headings of the birds that have participated in the formation are very close to each other. For the convenience of observation, we set the radius of the points representing the birds in polar coordinates as an arithmetic sequence. Thus, the correspondence of the bird flock in polar coordinates is that multiple points are approximately on a straight line. In addition, not all points can form a perfectly straight line when the simulation stops, which indicates that not all birds participate in SoS to form a formation. Although the flight formation does not entirely form when the simulation stops, there is still a trend of convergence overall compared with the initialization, which means the birds that have not yet joined the formation still have a chance to join the SoS.

In addition to modeling SoS types, we also found that for the bird flock SoS, the transition between different SoS types is of positive significance. When the simulation stops, the birds' headings in the acknowledged SoS are denser than in the collaborative SoS, as shown in Fig. 5(b) and Fig. 6(b), which means that the flight formation formed in the acknowledged SoS has more members participating. Furthermore, some birds have not participated in the formation in the acknowledged SoS when the simulation stops. The headings of these birds tend to be more towards the formation than in the collaborative SoS. Fig. 6(b) shows that almost all birds head at 0° to 90°, where the formation heading in the acknowledged SoS is. Conversely, some birds head at 180° to 270°, which is far from the direction of the formation in the collaborative SoS, as shown in Fig. 5(b). We realize the transition between the collaborative SoS and the acknowledged SoS in the bird flock SoS by enabling or disabling Rule 1 and Rule 4. Therefore, under the simulation environment parameters set in this paper, we use quantitative methods to demonstrate different SoS types and find that the acknowledged SoS has a better overall effect than the collaborative SoS for the bird flock SoS. That is to say, if a specific SoS practice has different SoS types, each SoS type may result in different emergent behaviors. We can obtain better SoS effectiveness by converting the existing SoS type to a more suitable SoS type. Furthermore, the transition between different SoS types helps migrate technical methods that have been successfully applied in one SoS type to other SoS types.

We also found other reasonable explanations for the transition between different SoS types. Holland [49] believed that there is competition between multiple rules for controlling the behaviors of agents, which results in a contradiction. Therefore, a way to resolve the contradiction is needed. He proposed that competition should base on experience, and the ability of a specific rule to win the competition should base on the past use of the rule. Thus, it is necessary to assign a strength to each rule. After a period of time, the strength will reflect the usefulness of the rule, and the process of modifying the strength based on experience is called credit assignment. Similarly, the relationship between the top-down authority of SoS and the bottom-up independence of CS appears to be competition. In directed/acknowledged SoS, the strength of the top-down authority of SoS is greater than the strength of bottom-up independence of CS in collaborative SoS. Thus, the transition between directed/acknowledged SoS and collaborative SoS means the change of this strength. In the bird flock SoS, there is a mechanism for every bird to circulate as the leading bird. According to the SoS types distinction approach proposed in this paper, when the middle bird of a formation switches to the leading bird, the SoS is regarded as an acknowledged SoS. However, suppose this mechanism is regarded as a spontaneous generation of the SoS, which means that the strength of the top-down SoS authority changes into the strength of the bottom-up independence of CS. In that case, it may be similar to the credit assignment mechanism [49], then the SoS can also be regarded as a collaborative SoS. In essence, the credit assignment is a kind of benefit that cannot be obtained immediately but is believed to be obtained in the future, which is similar to that a "latent potential" only after participating in an SoS in the future can be activated to a "realized potential".

8. Conclusions

This paper presents a goal-based approach that provides a mathematical baseline to translate the definition of SoS types into ABM and simulations. This approach constructs the theoretical models of CSs and SoS by introducing an additional type of function called "potential" that can only be activated by another CS's outputs instead of its resources, making the relationship between SoS characteristics and SoS types more explicit. Next, the influence of SoS characteristics on SoS goals and CS goals is analyzed to discuss all possible situations under authority. Then, a theory of SoS types embodied in the rigorous model by explaining the degree of authority quantitatively with the direction of change in goals is proposed to facilitate the definition of different SoS types. Finally, the feasibility of the proposed approach is illustrated by using a migrating waterfowl flock by an ABM simulation. The modeling approaches recognize the difference between collaborative SoS and acknowledged SoS by enabling or disabling rules controlling behaviors CSs.

The proposed approach is particularly well suited for SoSs without explicit authority but with explicit changes of SoS goals and CS goals. The primary value of the approach is providing a new perspective for understanding SoS theory-related aspects based on goals. The proposed concept of "potential" helps understand the difference between the CS in SoSE and the system in SE by enriching the content of SoS goals and CS goals. With that as a basis, better fundamental theories of SoS (e.g., meta-model [50,51] or ontology [52,53]) and methodologies of SoSE (e.g., mission engineering [54,55]) may be facilitated.

The SoS type may not be uniform [8], and the elements contained in the SoS exhibit characteristics of different SoS types [13]. We argue that some SoSs have only one dominant type. Also, some SoSs have multiple types, none of which are dominant. For an SoS with a dominant type, the explanation of SoS classification based on transitions of goal states can help engineers recognize the SoS type correctly and early in the analysis stage. Thus, engineers can avoid misclassification and select corresponding methodologies for a certain type of SoS in early SoS development. In some actual SoSs, such as swarm robots, the SoS may have alternative types. The goal-based approach proposed in this paper for modeling and simulating SoS types can help engineers compare the effectiveness among different SoS types, guiding the selection of appropriate SoS types when constructing an SoS.

The limitations of the approach proposed in this paper are as follows. The "potential" in the theoretical model is easy to detect beforehand in SoSs where the potential of each CS is similar such as migrating waterfowl flock, swarm robots, and unmanned aerial vehicles [56]. However, the potential may be hard to detect beforehand in some complex CSs. We suppose that an iterative modeling and simulation process may be required, where simulation plays a more significant role in the SoS environment than in the SE environment [57], but it still requires further effort. In addition, the guidance of SoS types for SoS design is only reflected by enabling or disabling some simple rules to control the behaviors of CSs. Thus, it is necessary to develop mechanisms such as a genetic algorithm [49] that may be used to generate effective rules for SoS design. Moreover, it is challenging to model virtual SoS examples using the goal-based theoretical approach because a virtual SoS does not have explicit SoS goals. Future work can be divided into two directions. First, look for more complex SoS examples to enrich the description and modeling of SoS types. Second, it will focus on guiding the SoS design methods for different types of SoS and how to guide the extension or migration of the methods or techniques applied in a specific type of SoS to other types of SoS.

References

[1] DOD. Systems engineering guide for systems of systems. Washington: US Department of Defense (DoD), 2008.

[2] MITRE. Systems engineering guide. https://www.mitre.org/sites/default/files/publications/se-guide-book-interactive.pdf.

[3] DELAURENTIS D A, CROSSLEY W A, MANE M. Taxonomy to guide systems-of-systems decision-making in air transportation problems. Journal of Aircraft, 2011, 48(3): 760–770.

[4] OKAMI S, KOHTAKE N. Transitional complexity of health information system of systems: managing by the engineering systems multiple-domain modeling approach. IEEE Systems Journal, 2017, 13(1): 952–963.

[5] DELAURENTIS D. Understanding transportation as a system-of-systems design problem. Proc. of the 43rd AIAA Aerospace Sciences Meeting and Exhibit, 2005: 123.

[6] STEGHOFER J P, ANDERS G, SIEFERT F, et al. A system of systems approach to the evolutionary transformation of power management systems. Proc. of the Informatik 2013—Workshop on “Smart Grids”, 2013: 1500−1515.

[7] JAMSHIDI M. System of systems engineering: innovations for the 21st century. Hoboken: John Wiley & Sons, 2008.

[8] NIELSEN C B, LARSEN P G, FITZGERALD J, et al. Systems of systems engineering: basic concepts, model-based techniques, and research directions. ACM Computing Surveys, 2015, 48(2): 1–41.

[9] MAIER M W. Architecting principles for systems-of-systems. Proc. of the INCOSE International Symposium, 1996, 6(1): 565–573.

[10] MAIER M W. Architecting principles for systems-of-systems. Systems Engineering, 1998, 1(4): 267–284.

[11] DAHMANN J S, BALDWIN K J. Understanding the current state of US defense systems of systems and the implications for systems engineering. Proc. of the 2nd Annual IEEE Systems Conference, 2008: 1−7.

[12] SILVA D A, BRAGA R A, VACCARE R T. Simulating systems-of-systems with agent-based modeling: a systematic literature review. IEEE Systems Journal, 2020, 14(3): 3609–3617.

[13] DAHMANN J S. Systems of systems characterization and types. https://www.sto.nato.int/publications/STO%20Educational%20Notes/STO-EN-SCI-276/EN-SCI-276-01.pdf.

[14] DAHMANN J. System of systems pain points. Proc. of the INCOSE International Symposium, 2014, 24(1): 108–121.

[15] COOK S C, PRATT J M. Typology dimensions for classifying SoSE problem spaces. Proc. of the 11th System of Systems Engineering Conference, 2016: 1−6.

[16] TEKINERDOGAN B. Multi-dimensional classification of system-of-systems. Proc. of the 14th Annual Conference System of Systems Engineering, 2019: 278−283.

[17] MOHSIN A, JANJUA N K, ISLAM S M, et al. Modeling approaches for system-of-systems dynamic architecture: overview, taxonomy and future prospects. Proc. of the 14th Annual Conference System of Systems Engineering, 2019: 49−56.

[18] ENDER T, LEURCK R F, WEAVER B, et al. Systems-of-systems analysis of ballistic missile defense architecture effectiveness through surrogate modeling and simulation. IEEE Systems Journal, 2010, 4(2): 156–166.

[19] BAEK Y M, MIHRET Z, SHIN Y J, et al. A modeling method for model-based analysis and design of a system-of-systems. Proc. of the 27th Asia-Pacific Software Engineering Conference, 2020: 336−345.

[20] DRIDI C E, BENZADRI Z, BELALA F. System of systems modelling: recent work review and a path forward. Proc. of the International Conference on Advanced Aspects of Software Engineering, 2020: 1−8.

[21] GE B, HIPEL K W, YANG K, et al. A novel executable modeling approach for system-of-systems architecture. IEEE Systems Journal, 2014, 8(1): 4–13.

[22] AXELSSON J. A refined terminology on system-of-systems substructure and constituent system states. Proc. of the IEEE Systems of Systems Conference, 2019: 31−36.

[23] BOARDMAN J, SAUSER B. System of systems-the meaning of of. Proc. of the International Conference on System of Systems Engineering, 2006: 118−123.

[24] BALDWIN W C, FELDER W N. Mathematical characterization of system-of-systems attributes. KAHLEN F J, FLUMERFELT S, ALVES A. Transdisciplinary Perspectives on complex systems. Berlin: Springer, 2017.

[25] BALDWIN W C, BEN-ZVI T, SAUSER B J. Formation of collaborative system of systems through belonging choice mechanisms. IEEE Trans. on Systems, Man, and Cybernetics—Part A: Systems and Humans, 2012, 42(4): 793–801.

[26] BALDWIN W C, SAUSER B. Modeling the characteristics of system of systems. Proc. of the International Conference on System of Systems Engineering, 2009: 1−6.

[27] MOKHTARPOUR B, STRACENER J. A conceptual methodology for selecting the preferred system of systems. IEEE Systems Journal, 2017, 11(4): 1928–1934.

[28] DARABI H R, MANSOURI M. The role of competition and collaboration in influencing the level of autonomy and belonging in system of systems. IEEE Systems Journal, 2013, 7(4): 520–527.

[29] FELDER W N, BALDWIN W C. Estimation of the belonging metric in a hypothetical system-of-systems. IEEE Systems Journal, 2019, 13(2): 1936–1944.

[30] BALDWIN W C, FELDER W N. Use of the belonging metric to inform architectural decisions in an air defense scenario. Procedia Computer Science, 2019, 153: 166–176.

[31] LANA C A, GUESSI M, ANTONINO P O, et al. A systematic identification of formal and semi-formal languages and techniques for software-intensive systems-of-systems requirements modeling. IEEE Systems Journal, 2019, 13(3): 2201–2212.

[32] CHERFA I, BELLOIR N, SADOU S, et al. Systems of systems: from mission definition to architecture description. Systems Engineering, 2019, 22(6): 437–454.

[33] SILVA E, BATISTA T, OQUENDO F. A mission-oriented approach for designing system-of-systems. Proc. of the 10th System of Systems Engineering Conference, 2015: 346−351.

[34] LANE J A, EPSTEIN D. What is a system of systems and why should I care? https://citeseerx.ist.psu.edu/doc/10.1.1.365.8502.

[35] GOROD A, SAUSER B, BOARDMAN J. System-of-systems engineering management: a review of modern history and a path forward. IEEE Systems Journal, 2008, 2(4): 484–499.

[36] BALDWIN W C, SAUSER B J, BOARDMAN J. Revisiting “The Meaning of Of” as a theory for collaborative system of systems. IEEE Systems Journal, 2017, 11(4): 1–12.

[37] SOYEZ J, MORVAN G, MERZOUKI R, et al. Multilevel agent-based modeling of system of systems. IEEE Systems Journal, 2017, 11(4): 2084–2095.

[38] SEO D, SHIN D, BAEK Y, et al. Modeling and verification for different types of system of systems using PRISM. Proc. of the 4th International Workshop on Software Engineering for Systems-of-Systems, 2016: 12−18.

[39] BALDWIN W C, SAUSER B, CLOUTIER R. Simulation approaches for system of systems: events-based versus agent based modeling. Procedia Computer Science, 2015, 44: 363–372.

[40] HESTER P T, TOLK A. Applying methods of the M&S spectrum for complex systems engineering. Proc. of the Spring Simulation Multiconference, 2010: 1−8.

[41] BERTALANFFY L V. An outline of general system theory. The British Journal for the Philosophy of Science, 1950, 1(2): 134–165.

[42] HOSKING M, SAHIN F. Testability of a swarm robot using a system of systems approach and discrete event simulation. Proc. of the 5th International Conference on System of Systems Engineering, 2010: 1−6.

[43] SAGE A P, CUPPAN C D. On the systems engineering and management of systems of systems and federations of systems. Information Knowledge Systems Management, 2001, 2(4): 325–345.

[44] AGARWAL S, PAPE L E, KILICAY-ERGIN N, et al. Multi-agent based architecture for acknowledged system of systems. Procedia Computer Science, 2014, 28: 1–10.

[45] GRIMM V, BERGER U, DEANGELIS D L, et al. The ODD protocol: a review and first update. Ecological Modelling, 2010, 221(23): 2760–2768.

[46] GRIMM V, BERGER U, BASTIANSEN F, et al. A standard protocol for describing individual-based and agent-based models. Ecological Modelling, 2006, 198(1): 115–126.

[47] NATHAN A, BARBOSA V C. V-like formations in flocks of artificial birds. Artificial Life, 2008, 14(2): 179–188.

[48] WILENSKY U. NetLogo: center for connected learning and computer-based modeling. http://ccl.northwestern.edu/netlogo/.

[49] HOLLAND J H. Hidden order: how adaptation builds complexity. New York: Addison Wesley Longman Publishing Co., Inc., 1996.

[50] BAEK Y M, SONG J, SHIN Y J, et al. A meta-model for representing system-of-systems ontologies. Proc. of the 6th International Workshop on Software Engineering for Systems-of-Systems, 2018: 1−7.

[51] DRIDI C E, BENZADRI Z, BELALA F. System of systems engineering: meta-modelling perspective. Proc. of the 15th International Conference of System of Systems Engineering, 2020: 135−144.

[52] FAKHFAKH S, JANKOVIC M, HEIN A M, et al. Proposition of an ontology to support product service systems of systems engineering. Systems Engineering, 2021, 24(5): 293–306.

[53] FRANZEN L K, STAACK I, KRUS P, et al. A breakdown of system of systems needs using architecture frameworks, ontologies and description logic reasoning. Aerospace, 2021, 8: 118.

[54] SOUSA-POZA A. Mission engineering. International Journal of System of Systems Engineering, 2015, 6(3): 161–185.

[55] BEERY P, PAULO E. Application of model-based systems engineering concepts to support mission engineering. Systems, 2019, 7(3): 44.

[56] OQUENDO F. Exogenously describing architectural emergent behaviors of systems-of-systems with SosADL. Proc. of the 13th Annual Conference on System of Systems Engineering, 2018: 268−275.

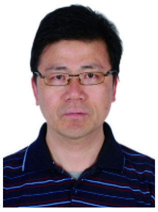
[57] HALLO L, PAYNE B, GOROD A. Model-based approach to system of systems engineering: reevaluating the role of simulation. Proc. of the 14th Annual Conference System of Systems Engineering, 2019: 266−271.

Biographies

FENG Yimin was born in 1995. He received his B.E. degree in computer science from Northwestern Polytechnical University, Xi'an, China, in 2017. He is currently pursuing his Ph.D. degree in computer science at the State Key Laboratory of Computer Aided Design and Computer Graphics, Zhejiang University. His research interests include system-of-systems engineering, model-based system engineering, and computer-aided design.
E-mail: yiminfengmy@zju.edu.cn

ZHOU Chenchu was born in 1990. He received his master's degree from Beijing University of Aeronautics and Astronautics in 2015. He is currently an engineer at Xi'an Aerospace Propulsion Institute. His research interests include engine dynamics simulation, aerospace propulsion theory and engineering.
E-mail: zhouchenchu@126.com

ZOU Qiang was born in 1990. He received his Ph.D. degree in mechanical engineering from University of British Columbia, Vancouver, Canada, in 2019. He is currently an associate professor at the State Key Laboratory of Computer Aided Design and Computer Graphics and the College of Computer Science, Zhejiang University. His research interests include design modeling and manufacturing simulation.
E-mail: qiangzou@cad.zju.edu.cn



LIU Yusheng was born in 1970. He received his B.E. degree in mechanical engineering and Ph.D. degree in computer science from Zhejiang University, Hangzhou, Zhejiang, China, in 1995 and 2000, respectively. He is currently a professor and a researcher at the State Key Laboratory of Computer Aided Design and Computer Graphics and the College of Computer Science, Zhejiang University. His research interests include model based system engineering, computer aided design and manufacturing, product innovation design, and model retrieval.
E-mail: ysliu@cad.zju.edu.cn

LYU Jiyuan was born in 1984. He received his B.E. and Ph.D. degrees from Northwestern Polytechnical University in 2008 and 2012, respectively. He is currently an engineer at China Astronaut Research and Trainning Center, Beijing, China. His research interests are guidance and control engineering, manned space mission analysis and simulation.
E-mail: lvjiyuan@139.com

WU Xinfeng was born in 1983. He received his Ph.D. degree from National University of Defense Technology. He is currently an engineer at China Astronaut Research and Trainning Center, Beijing, China. His research interests include aircraft design engineering, and manned space mission analysis.
E-mail: wxf.134@163.com