



## Enhancement Belief Function with Ontology to Solve the Conflict for the System of Systems

**Hebatullah R. Younes<sup>1\*</sup>**

**Ahmed Sharaf Eldin Ahmed<sup>2,3</sup>**

**Eman K. Elsayed<sup>4</sup>**

<sup>1</sup>*Computer Science Department, Sinai University, Arish, Egypt*

<sup>2</sup>*Information Technology Department, Sinai University, Arish, Egypt*

<sup>3</sup>*Information System Department, Helwan University, Cairo, Egypt*

<sup>4</sup>*Mathematical and Computer Science Department, Al-Azhar University, Cairo, Egypt*

\* Corresponding author's Email: heba.alrashed@gmail.com

---

**Abstract:** System of Systems (SoS) is a huge system consisting of small stand-alone systems to offer unique abilities. In such complex construction, managing achievement of SoS objectives along with those of its Component Systems (CS) is a challenge. Whereas the integration of CS may cause an extended impact issue called conflict. In this research, the principles of basic belief function were used to measure conflict in the SoS. This is intended to detect conflicts between CS in an attempt to resolve it. Moreover, we have based on the ontology in the SoS integration. In addition to the identification of relationships between the CS semantically. We have proposed a method to resolve conflicts called Ontology Belief Function System of Systems (OBFSoS). As a homogenous SoS, we apply the OBFSoS on Real-time Strategy (RTS) game. We depend on the use of a learning agent. It reduces conflicts and saves time due to attempts. For execution, we got the result of 10 attempts with about 30 systems (players) in the SoS. It giving a good decision in a short time with an accuracy of approximately 87%.

**Keywords:** System of systems (SoS), Belief function, Conflict, Ontology, Decision, Real-time strategy (RTS) game.

---

### 1. Introduction

In the last years, there are small systems that jointly formed into a larger complex system as a system of systems (SoS) to offer unique features and abilities [1, 2]. Usually, the SoS capacity is greater than the total capacity of its component parts [3]. It connects the different parts with the whole to solve large-scale problems. We can represent the plane as an example of a complex and extensive system of systems. Many systems, operate various parts of the plane. The plane flies only when all its systems work side by side. It doesn't fly if the systems operate independently of each other [4]. Smart cities are another example. They typically encompass several of distributed systems. These systems are integrated into complex relationships. This including interaction

with other systems towards providing new functionalities [5].

However, depending on the SoS structure, we have many challenges. Lack of evident pattern and big data are some of them. But one of the most important challenges is the conflict. Discord arising within a group of component systems in the SoS has called the conflict. Component Systems in the SoS construction may present conflicting individual targets or decisions among themselves. We focus on the conflict in decision-making for the SoS results from its components.

In order to successfully develop any system, it needs to take into account its targets. In the SoS, each component system has its own goals, interests, and decisions. This makes it difficult for the SoS to achieve all these goals together without conflict. The conflict has several situations in the SoS. Such as

Conflict due to the connecting pattern, classification, Security, and Decision.

For the SoS, Ontology can organize how such systems may be grouped. It is an important method in the integration process. In addition to this, there is a function that clarifies how to solve the conflict between evidence. It is called Belief function. We will use this function with ontology to handle the conflict in the SoS.

Different conflict problems had been solved. The produced solutions for belief functions to detect the conflict was successful in most searches. But they did not implement on the SoS. Some attempts that implement on the SoS to solve the conflict in design. For a decision in the SoS depends on its management. Management issues result from the overlapping authority over the decision. This means that security is a part of the SoS.

As we mentioned, if we need to get an optimal decision between systems in the SoS, we need to manage the SoS. Decision-making represents a mapping from state and action. For this, in our approach, we try to use the belief function method to handle the conflict. Here we will get the benefit of this method on the SoS.

There are two types of SoS; homogenous and heterogeneous system. The Real-Time Strategy games are classified as homogenous SoS. RTS (Real-Time Strategy) game [6] are based on actions and reactions. The RTS game needs more than one player to play together through the network. As is known in the system definition. It is necessary to have the hardware, software, and communication among them to create system enables people to use. Hardware is the tool the player needs to play with the game software. The connection between them can be through the network in the online version of the game. Since each player has his own tools to be a specific system with its details.

In this paper, we propose a method to handle the conflict in the decision in the SoS. The proposed method OBFSoS hybrid the enhancement belief function with Ontology. Where using ontology is to detect if some conflict resulted from similarity as we will clarify in the next section. We will implement the homogeneous SoS.

This paper is organized as follows. Section 2 covers the main concepts in the background as System of Systems, its issues concentrating on conflict problem. Also, give a brief of Ontology and belief function that we will use in our proposed method. In section 3, the proposed method Ontology Belief Function System of Systems (OBFSoS) is described in detail. Section 4 comprises the implementation of the OBFSoS on two case studies

on RTS game and another. In section 5, the simulated results and discussion are introduced. Finally, section 7 outlines our conclusions.

## 2. Background

### 2.1 System of systems (SoS)

The system is defined as a combination of hardware, software, people and communications between them. This group cooperated together to solve a problem or achieve a specific goal. For System of Systems, we get a group of systems together to create a system out of them (SoS). These systems are connected to each other to accomplish the SoS task. In addition to that, systems can accomplish their own tasks.

About SoS we can imagine that there are large-scale systems. This includes sophisticated interactive component systems that evolve in a dynamic environment. Based on mainstream modelling of the System of Systems; J.-B et al., In 2015 [7] provides a general description of the complexity and control of the SoS. The researchers built a Multilevel Simulation agent-based model that organizes SoS. In accordance with the ability of subsystems or changing targets. The researchers in this paper take into consideration that subsystems in SoS can be evolved over time. This is an important method. But although system problems such as the conflict between subsystems in the SoS have not been addressed.

### 2.2 Ontology

It can divide systems, according to similarities and differences. So that coordination between systems in the SoS may require Ontology [8]. Using Ontology is due to the development process. It makes the common ontology extends across multiple areas of the SoS. This method can be critical, if not pivotal [9]. That is for the success of the SoS interoperability efforts which are inherently multi-disciplinary and collaborative.

In the SoS we need to use ontology to the process of the Interoperability among systems. More broadly the rules for interoperability of systems include a means of the SoS connecting. Depending on experience and knowledge from subsystems in the SoS [10]. The ontological technology and synthetic analysis are analysed as elements, relationships, and field knowledge. That would influence the SoS formation, utilization, evaluation, and model relationship factors. A common language, a common

grammar and a common set of meanings in given contexts also included.

In addition to these operational timelines that ensure that information is never delayed [8,11]. Individual systems of the SoS are designed separately in advance. So that interoperability requirements are not necessarily met without new interfaces [8].

Focusing on grammatical language requirements. The preferred approach is for all systems to interact through a common interface. This interface can be created using multiple layers of ontologies. There are some requirements for the exchange of information required for interoperability, this need ontology.

G. Abdalla et al., In 2015 [12] introduced a Systematic Literature Review (SLR) to determine how to apply the representation of knowledge to the SoS. They use ontology as a term that describes and represents an area of knowledge. The researchers did not concern with the SoS concepts or terminology formalization. This could be considered a research gap to be explored in the future.

The Service-oriented modelling method for analysing a large-scale complex system such as the SoS requirements was the research which Osmundson et al., In 2012 [13]. They proposed a reusable method of service-oriented and ontology-based requirements modelling to develop SoS architecture.

### 2.3 Belief function

It is used to reduce the mass conflict, according to the reliability estimate. These functions will assist in the suggested algorithm for resolving the SoS conflict issues discussed in this paper.

For Belief function theory, conflict is an important concept. That is because the combination of sources causes conflict. The theory of belief functions was first introduced by [14]. In 2006 W. Liu determine when there are fundamental conflicts in beliefs [15]. He suggested the mathematical theory of evidence. The theory relies on handling the conflict that results from the integration between information [16]. The unreliability of sources allows the conflict. Sources that do not give information about the same situation also allow the conflict.

We will clarify how we use belief function in determining the conflict but for the SoS. Not as it defined for information or sources only.

There were many attempts from researchers as in 2017 [17]. Martin discusses how to manage the conflict based on the assumption that the conflict comes from. He depends on the belief function. Many methods for the conflict measures to estimate the reliability of information sources were introduced.

Depending on how to manage the conflict using belief functions.

### 2.4 The conflict types in the SoS

In order for that conflict is the main issue with special dimensions, we adopt this issue in particular. The conflict effect on other SoS issues also. Then SoS There are many types of conflicts that we will show in the next subtitles.

#### 2.4.1. The conflict due to the connecting pattern

Software engineering methods have the main contribution. They help in the integration between constitutive systems of SoS [18]. Most studies describe individuals. In addition to teams that have worked in isolation to develop solutions. That used to specific problems without widespread adoption of a form of integration. Systems integration makes an important ability by merging subcomponents. It is the logical next step between design and development. Testing, verification, validation, and deployment also. Integrate subsystems together is an important issue to prevent conflict.

The conflict may occur from the beginning due to the faulty way of the integration. We will try to choose the best way to integrate subsystems to build the SoS. We have more than one method. The complexity of constructing the SoS due to the inherent complexity of these systems. There are various kinds of complexity that arise in software SoS. All systems are consisting of parts (elements) that have relationships between them. As an example, program parts may be objects and parts of each object may be constants, variables, and methods. Relationships include Calls (method A calls method B), Inherits-from (object X inherits the methods and attributes of object Y) and Part of (method A is part of object X). The complexity of any system depends on the number and the types of relationships between its elements.

Connecting systems together in the SoS contains some agreed models as Star, Vertical and Horizontal. Each model has its advantages and disadvantages. That may measure by cost, time or other things according to each organization direction.

The SoS is formed by combining complicated independent systems. This will increase the complication of the SoS to at least one more order over its constituent systems. This means that problems in the SoS environment are harder to handle than in the component system environment. T. A. de Melo Novaes Viana In 2017 [19] in her research presents an overview of the SoS framework. That supports conflict management in the SoS. The

management of conflicting requirements in the proposed framework includes three main steps for conflict identification, the conflict diagnosis, and resolution based on the use of a utility function. She had built a prototype of the framework for the SoS designed to support food security Feed Me. In 2009 A. Gorod et.al., [20] give the SoS framework management. They try to reduce the complexity of management and increase operational effectiveness for the SoS.

Merging subsystems into the SoS structure is the first step we take to prevent conflicts. Choosing the best method depends on the number of subsystems in the SoS formation. Time and cost are also an important element. The method of integrating these systems together without conflicts. All that we will try to achieve in addition to keep the time within our graphs also.

#### 2.4.2. The conflict due to the SoS classification

Conflict in System of Systems can depend on the basic classification of the SoS. We will present the classification scheme for SoS. That's based on the governance and management complexity of the SoS.

Directed SoS belongs to a single organization. It is developed through the integration of the systems. System components can be might be autonomously overseen [21]. The problem can be solved between different system through the administrative components of it. When the subsystems belonged to the same organization. The conflict may not occur because of the management of the organization to these systems. Also, of knowing the priority of the system management.

Collaborative SoS is systems where there is no central authority. This prioritizes management and conflict resolution. Typically, system elements are owned and controlled by different organizations. However, all of the organizations concerned recognize the mutual benefits of the common governance of the system. Therefore, they usually set up a voluntary administrative body that makes decisions on the system [21]. Some of its examples are a bus, rail, and air transport providers agree to link their systems. They provide passengers with up-to-date information.

This type of SoS is close to the idea of conflict control. All systems are linked to each other to make a decision. This decision is special to solve another problem. This will be one of the types of conflict that our paper tries to handle (decision conflict).

Virtual SoS. [21] do not have centralized governance. It depends on the published interfaces that may change. These systems have a very high

degree of both managerial and governance complexities. An example of it is an automated high-speed algorithmic trading system. These systems of different companies automatically buy and sell shares from each other.

The short paper from E. Honour In 2013 [1] introduced key ideas about verification and validation in the SoS environment. With persistent SoS development, administration of the two requirements and system configurations is an issue. Often, it is difficult to keep up a valid set of requirements for the SoS because of the progressing changes in the constitutive systems. Validation recognizes the essentially subjective nature of the conflicting. Transformative objectives and permitting the SoS configuration to adapt after some time as required.

Determining the number of systems. Also, the followed modifications can't limit the place of the conflict. As it may be from the systems that left. This may lose time. A new problem may occur also with other systems in the SoS.

#### 2.4.3. The conflict due to security in SoS

"The risk of information leaks is more important than the risk of services available" This expression follows in information security. That means that security has primary importance for any system demands that. In information security, some parts of a system may work offline. That is because security is very important. The security may keep the system offline for the encryption system. As in the encryption system, any system can be penetrated, regardless of the degree of insurance. Hacking by the system owner to handle any problem with it. Whether the problem is a fingerprint, a password that has been damaged, removed, or something else. May also have the eye, sound passwords and the person no longer exist. In such cases, the use of each code analysis may handle the problem. Measuring the strength of our security system. For example, breaking or removing the password after two attempts, five or ten attempts. We can say it is strong, but ultimately it is removed. The perfect solution is to change the password each period well before in advance to break it.

From this small interpretation. We can conclude that security is an open issue. That represents a conflict with more problems. From the SoS perspective, security is a difficult problem. As we have said in some systems such as Military intelligence services. Citizen data in the Civil Registry, the Passport and Immigration Authority also. Control in some sensitive cases in the health system, especially in hospitals and others. It is not

possible to allow the validity of access to the patient's data for anyone entering the system. For example, if we assume that we have two systems of the SoS structure. System X grants permission to someone to enter systems in the SoS and System Y granting this permission only to persons with certain specifications. This situation would make conflict. So, to maintain security in the SoS. And to permit the implementation of system models. The idea of providing communication structures must be followed as an integral part of the track information system specifications [22]. The SoS architecture must assemble the functions of the connection functions as a primary specification structure. In order to create a graphical flow of control over the collaborative assembly.

How to simulate the System of Systems model that prevents the attack on an organization? This question was asked by Ormrod et.al., In 2015 [23] to try to respond it in a practical way. The researchers provide a specific data model depending on ontology for the SoS subsystems. They simulate the SoS model that prevents the attack on an organization. Especially if the organization is military. They introduced the concept of independent ontologies designed the data model.

Through all of this, but they did not model using the ontology to avoid the security problems caused by the conflicting.

Description of an actionable engineering framework for security engineering of the SoS is the paper created by Dahmann et.al., In 2013 [24]. The framework is expected as a tool to assess security risks on command and control. In addition to other critical capabilities or missions based on an analysis of the contributing systems. In their approach, the analysis and corrective actions can be carried out to enable rapid response on an emergency basis. This is when critical risks emerge. The framework identifies critical elements in social service on the basis of impact analysis of mission objectives.

Finally, the dynamic defense environment is described by a variety of changing dangers. By disabling the ability of adversaries to defend and maintain an effective defense. This danger environment has increased awareness of system weaknesses. It made security a major concern for operations. In this direction, Baldwin et.al., In 2011 [25] did a study of the role of security in the SoS. Moving in the SoS environment poses new challenges to system engineering in general. In addition to specific efforts of the Department of Defense to engineer secure military capacity. Their research examined the approach of the United States

defense to security and the challenges posed by the SoS.

#### 2.4.4. The conflict due to decision in SoS

In some cases, we don't need to reduce conflicts [17]. We may use the conflict measure as an indicator of the inconsistency of the systems. We explained this in connecting or integration subsystems in the SoS. Conflicting data may use also to test our SoS for its issues. On the other hand, we need to resolve this conflict problem. So that we can reach the optimum decision of SoS. In this case, the conflict may appear in various areas. It depends on the systematic study and the region. When the conflict records a high value, then SoS subsystems give a different decision. We need to make the decision on true probabilities. For that, we need to manage the conflict to differentiate between production.

We choose this kind of conflict, especially to study (handle). We can say that it is under the name "Decision support system". Although other conflict types are important, decision-making in the SoS is a problem affecting all other issues. The SoS coordinating work between the systems without having a clear decision is so difficult. Also, the SoS handling the problem insecurity (for example) without a decision. The decision that maintains the security for all the SoS systems.

Because our goal is to handle the conflict in the decision between systems in SoS. So, we need a strategy to obtain the optimum decision in SoS. From here we can say that we are at the beginning of the idea. In this strategy, we need some mathematical equations. So that we can set the dimensions of the problem as we will show in the following sections.

Fang et al., in 2019 [26] proposed a method of effective optimization of the system of systems decision. The decision-making process is in the allocation of organizational, regulatory and decision-making costs for the allocation of the airline fleet ensures. Independent decisions of each airline can approach an effective global solution in an ideal centralized state as air transportation systems.

In 2018 [27], Langa regarded the conflict as an organizing element in any organization. That is because there are many differences between cases, procedures, and decisions. To overcome this, the administration needs to determine sources of

conflict that may affect its priorities. The study gives importance to solving the optimal conflict within the school organization.

Nikas et.al., In 2018 [28] have conducted a comprehensive review. That is for the contribution of multi-standard analytical frameworks. In order to

climate policy support. The researcher’s particular emphasis on their collective decision-making capabilities. The drawing on the urgency of involving many stakeholders in the climate policy-making process.

Decision-making in the SoS is not clarified in a clear way for the researcher. So that in our approach we will produce a solution to get a decision for the SoS.

### 3. Ontology belief function system of systems method (OBFSoS)

#### 3.1 Handling belief function to be suitable for SoS

Let  $\Theta = \{\Theta_1, \dots, \Theta_n\}$  be a frame of exhaustive hypotheses. A basic belief assignment (BBA)  $m$  is the mapping from elements of the power set  $2^\Theta$  (composed by all the disjunctions of  $\Theta$ ) onto  $[0, 1]$  such that:

$$\begin{cases} \sum_{X \subseteq \Theta} m(X) = 1 \\ m(\emptyset) = 0 \end{cases} \quad (1)$$

This means that System mass  $X$  represent SoS. Its default has not any conflict so that  $m(\emptyset) = 0$ . Meta-knowledge can be taken into account using the discounting operation  $\alpha$ .

$$\begin{cases} m^\alpha(B) = (1 - \alpha) \times m(B) & \forall B \subseteq \Theta \\ m^\alpha(\emptyset) = (1 - \alpha) \times m(\emptyset) + \alpha \end{cases} \quad (2)$$

Assuming that an information source represented with System ( $B$ ) has a reliability rate equal to  $1 - \alpha$  where  $0 \leq \alpha \leq 1$ . A discount rate  $\alpha$  equal to 1 means that the source is not reliable. That’s mean that the piece of information that provided cannot be taken as a true decision. On the contrary, a null discount rate indicates that the source is fully reliable. Also, the piece of information it provides is entirely true. Thanks to the discounting, an unreliable source’s BBA is transformed into a function assigning a larger mass to  $\emptyset$ . Now we know that  $\alpha$  may have any value between 0 and 1. This determine clarifies the credibility of the information. The lower the value is of  $\alpha$ , the more likely these concepts are to the truth. It can represent the decision-making process for the SoS. If the value resulted value were equals for all systems, it may make the conflict. In the following, we present several combination operators allowing systems to merge.

In Eq. (3)  $B$  and  $C$  are two systems belongs to SoS components  $X$ . Each system of them has its own

mass  $m$ . That is indicate to its components, relationships and, decisions.

$$m_\cap(X) = \sum_{\substack{B \cap C = \emptyset \\ \subseteq X}} m_1(B) \times m_2(C) \quad \forall B, C \quad (3)$$

To integrate the conflict management approach. That redistributes the generated conflictual mass. It relies on the rule as follows:

$$m_\cap(X) = \frac{1}{1 - K} \sum_{\substack{B \cap C = \emptyset \\ \subseteq X}} m_1(B) \times m_2(C) \quad \forall B, C \quad (4)$$

Where  $K$  representing the conflict between  $m_1$  and  $m_2$  and  $m(\emptyset)$  represent the conflictual mass between a two system  $B$  and  $C$  in SoS mass  $X$ .

$$K = \sum_{B \cap C = \emptyset} m_1(B) \times m_2(C) = m_\cap(\emptyset) \quad (5)$$

The conflictual mass  $m_\cap(\emptyset)$  on a set of positions  $P$ , that is mean that if  $A$  is a conflict. So that we need to check its position in its system. Positions  $P$  are according to a weighting factor  $W(A, m) (A \subseteq P)$  with  $\{m_1, \dots, m_j, \dots, m_j\}$  The final mass after combination [29, 30],  $m^c$  is part of the conflicting mass and can be written as follows:

$$\begin{cases} m^c(A) = W(A, m) \cdot m_\cap(\emptyset) & \forall A \subseteq P \\ m^c(A) = 0 & otherwise \end{cases} \quad (6)$$

Such that:

$$\sum_{A \subseteq P} W(A, m) = 1 \quad (7)$$

This framework represents the greatest representation of any conflict management approach. It not only provides the largest conflict management framework. It also the possibility of allocating the redistribution of empty set mass using weighting factors.

The probability transformation  $Pr$  is generally considered as a good basis for a decision rule. It considers even the composite hypothesis in its treatment, formally:

$$BetP(H_n) = \sum_{\substack{A \subseteq \Theta \\ \in \Theta}} \frac{|Pr_n \cap A|}{|A|} \times m(A) \quad \forall Pr_n \quad (8)$$

After clarifying the mathematical equations. We use to extract the conflicting weight and the probabilities of determining the decision. We are trying to convert these equations into an algorithm to be able to apply them it on SoS.

### 3.2 The proposed method OBFSoS

During our study of the SoS, we faced some issues and challenges. That is, which could hinder SoS from performing its main task. We found it is necessary to provide a solution. We, therefore, proposed a method that produces a solution. It helps to handle the conflict between decisions in the SoS. We choose this issue, particularly because of its importance. It enables the SoS to achieve its objectives of giving the best decision to large organizations. All other issues are related to the conflict anyway. We ensure that depending on our study. Studying show that the SoS can't work properly if there is a conflict between systems. Even if it works, it may cause problems in security or any critical case.

We give our method the name which holding its objective the OBFSoS. The OBFSoS referred to the full expression (Ontology Belief Function System of Systems). In the OBFSoS method, we offer a solution depends on some of the equations of belief function. In principle, we will make discount factors as inputs. These factors are which our OBFSoS depends on its values. That's for the decision-making to be followed by the SoS. A discount rate  $\alpha$  determines if SoS is in conflict or not. We clarified that in the explanation of belief function's Eq. (2). Factor  $\alpha$  when equal 1 depending on the equation in the OBFSoS method. That's mean a conflict and the non-real information is given as a resolution in this case. Conversely, 0 means that your SoS information is correct.

The conflict mass between the two systems  $s_1$  and  $s_2$  represents the value of  $k$ . We can present it with  $s(\emptyset)$  in OBFSoS method.  $k$  identifies the conflict systems (data or decisions) in the SoS. Factor  $W$  is the weight of each system. This is the factor which proposed in OBFSoS method to be depended on it on results. We proposed an equation to calculate the weight  $W$  of the systems whose decisions are conflicted together. Each system in the SoS has its own data collected. This used to give the real decision at the right time for it. We calculate the weight factor, according to data for each system. The weight will be the result of the number of the focal data points for the system. That used to produce the decision at a given time. We will divide it by the total number of focal-data.

It used to issue this decision from all conflicting systems. We rely on the weight because when a system depends on more data to make a decision, it means that it examines more data for the correct decision. You might think that the system might have more data, but all of it is wrong. This is the main cause to use the belief function theory. Also, rely on getting the  $\alpha$  value to verify the reliability of the data for each system first. The condition here is to get the decision of the system with the largest weight value. If there are equal values for the weight of more than one system. In this case, we have two options, is to merge the decisions together or leave the decision to the user. To determine the optimum state, we will try to check the rules we follow for SoS.

Here the role of fuzzy appears, depending on its logical properties. We will use the fuzzy in our approach just as 0 (false) or 1 (true) more clearly on and off for the system. We can do this in the case when our systems, decision weights were equal. Before merging the decisions or granting the user permission to choose. We will prevent some systems from appearing in the decision results. We relied on the idea of "The more times of the conflict for a system with the rest of the group of other systems, the less is reliable it is". So, we use fuzzy to control this process. Measure the number of times the system occurs in the conflict circle. We will use the time to detect that, for example, we may do it every 5 minutes.

The OBFSoS method output variable  $X_{ij}$  obtaining the optimum decision the SoS reached for it. We also need to note that we will order the conflict issue with the order of priority. Priority according to SoS requirements, we may need to use fuzzy also. For example, we will give the first priority of having the decision to the decisions of authorizing. Some users need to enter as admins in different types of systems that may cause a security problem. Therefore, some systems put the security as its first concern. Others systems the different priority order according to different objectives.

The output of the SoS may be any of the different types of file. The result may be in pdf type, word, alert message. Also, it may be an image as needed. In the algorithm (Algorithm 1) we write pseudocodes to explain the proposed method the OBFSoS in few steps.

---

**Algorithm 1** Ontology Belief Function System of Systems (OBFSoS)

---

- 
1. **Input**  $s_i (s_1, s_2, \dots, s_n) (i = 1, 2, \dots, n)$  // Systems in SoS
  2. Convert  $Os_i (Os_1, Os_2, \dots, Os_n) (i = 1, 2, \dots, n)$  // Ontologies
  3. Merge  $Os_i$  to  $OSoS$  // System of Systems Ontology
  4.  $\theta_j = (\theta_1, \theta_2, \dots, \theta_m) (j = 1, 2, \dots, m)$  // decisions of systems
  5. Calculate discount factors  $\alpha, k$  and  $w$
  6. measure initial state  $Os(\theta) = 1$
  7.  $s(\emptyset): k \leftarrow 0$  // initialize conflict mass
  8. start OBFSoS decision evaluation  $X_{ij} \leftarrow 0, s(\emptyset) \leftarrow 0$
  9.  $Os^\alpha(\theta) = (1 - \alpha) \times Os(\theta) + \alpha$
  10. if  $\alpha = 0$
  11.  $X_{ij} \leftarrow \theta_j$
  12. else
  13. **for**  $i = 2, 3, 4, \dots, n$  **do**
  14. for  $j = 1, 2, 3, \dots, m$  **do**
  15.  $k = Os_{i1}(\theta_{j1}) \times Os_{i2}(\theta_{j2}) = s(\emptyset) \quad \theta_{j1} \cap \theta_{j2} = \emptyset$
  16. Conflict mapping from  $OSoS$
  17.  $Conf(i, p) = k(s_i, s_p)$
  18. Calculate weight
  19.  $w_i = \frac{\text{no. focal data}}{\text{total number of focal data}}$
  20.  $w_{i1} > w_{i2} \dots w_{i1} = w_{i2}$
  21. **until**
  22.  $X_{ij} \leftarrow \theta_{j1} \dots X_{ij} \leftarrow \theta_{j1} \bowtie \theta_{j2}$
  23. **end for**
  24. **end for**
  25. **Output**  $X_{ij}$
- 

#### 4. Case study: applying OBFSoS method on RTS game

The OBFSoS was implemented using Java and packaged to the Glest RTS game engine. Converting into OWL Ontologies for all systems. In addition to merging the OWL ontologies using Protégé and its library to generate the universal Ontology for SoS.

##### 4.1 RTS game homogeneous SoS

Homogeneous means the systems of the SoS are using the same components. Also, do the same task. But here each system is in a different place. We choose an RTS (Real-Time Strategy) game which based on actions. Glest [6] is a free 3D RTS game. Where you control the armies of two different factions Tech and Magic. The tech which is mainly composed of warriors and mechanical devices. Magic, that prefers mages and summoned creatures in the battlefield. The game version needs more players to play together through the network. As is known in the system definition. It is necessary to have the

hardware, software, and communication among them to create system enables people to use. Hardware is the tool the player needs to play with the game software. The connection between them can be through the network in the online version of the game. Since each player has his own tools to be a specific system with its details. We can consider that players in the game are the proper structure for SoS.

We rely on the implementation to convert Systems into ontology. So that we can implement the method as we need to use a programming language such as Java. The results will illustrate the importance of ontology.

Actions in the game (Worker Harvest, Return Base, Massive Attack, Add Tasks, Produce Resource Producer, Build One Farm, Produce, Build, Upgrade, and Repair) represent data streams. Each set of actions belongs to a class, for example, the class (Units) includes [Build One Farm - Build]. Also, (Kills) [Massive Attack - Add Tasks] and class (Resources) have [Produce Resource Producer - Return Base]. In Fig. 5, the player (S1) has classes Kills, Units, and Resources; each class collects score depending on his actions. Agent (Sn) also has classes Kills, Units, and Resources. It is impossible to deal with classes with the same name without conflicts. Conflicts here can appear between SoS classes that have the same name with different players. Classes are similar in name with different information (details). When SoS tries to give a decision about who is the winning player. Or to be able to select from the two classes. In this case, the OBFSoS method first shows the similarity between the classes then yields the result.

This case is a parallel case where systems are arranged here to have their results specific to each player (system) then compare the results. The ontology here is useful because it depends on the semantic comparison. That help in distinguishes between the meanings of the classes of the same name and content with a different owner.

As part of the SoS, each player has his own strategy, scenario, usage cases, and classes. We convert these scenarios into the ontology. In addition to that, we implement the method. For Fig. 1, we show one of the conflicts that appear after ontology conversion between the two players. We named them as Player and Agent. We find that there is a conflict between the two classes that record score for the two players. The player has the same score as Agent. This cause a conflict to decide who wins the game. Using the semantics in ontology we can determine which score is associated with either player. We connect the actions of each player to score therefore we can



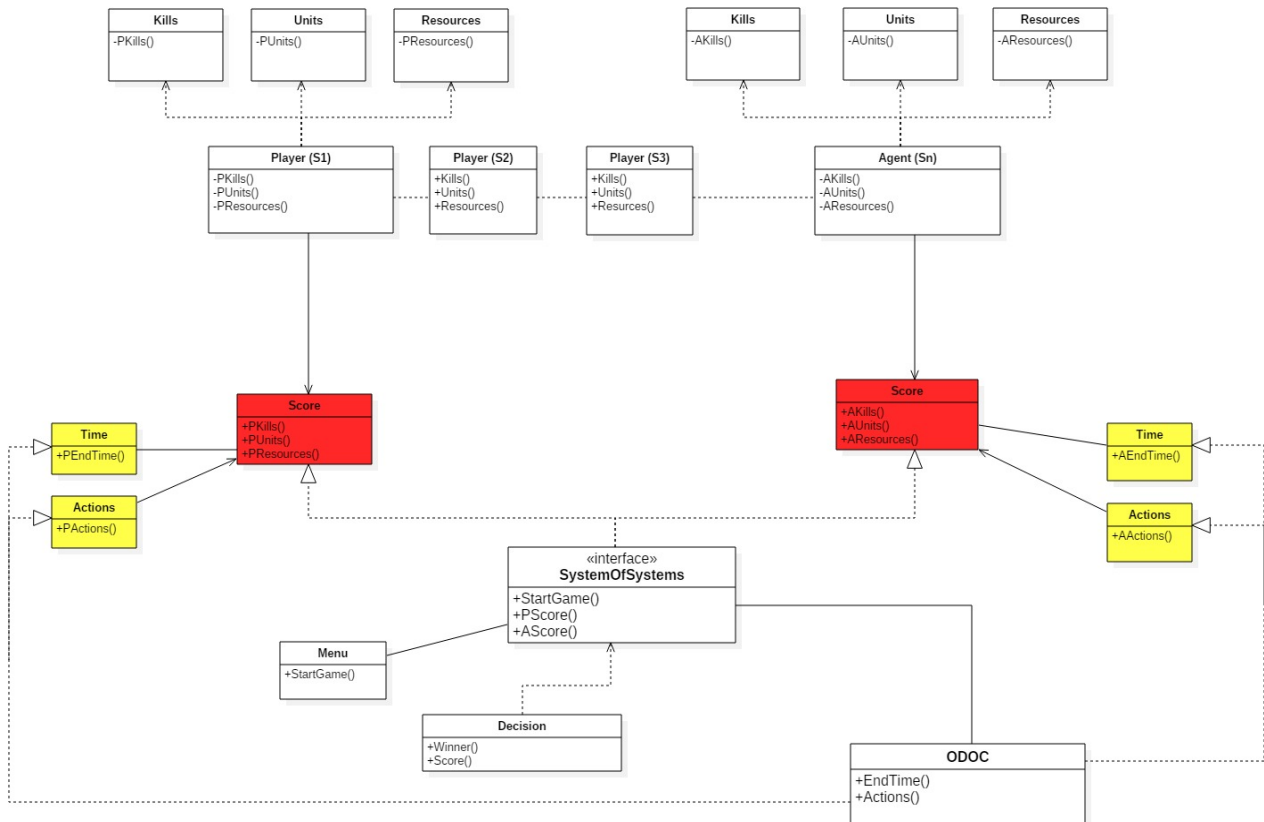


Figure. 1 RTS game homogenous SoS

determine the winner. Actions, as we say in the game, represents data streams and in the OBFSoS method. We need to be sure about (check) resource (data). Priority actions are arranged. So that the actions in categories (Kills, Units, Resources) can be ordered by requesting actions for each player in the same case. Time plays an important category also if the action gives the same order for such RTS games. Using a smaller number of actions with the same score gives the player the right to win the opponent.

### 5. Analysis of results and discussion

In this section, we will analyze the implementation of the OBFSoS method on RTS game as a homogenous SoS. The overall performance of 10 different attempts. That generated for the selected part of the SoS in the RTS game is calculated. The implementation is shown in Table 1. In this part of the SoS, we have about 30 systems (players). In order to calculate the number of conflicts between systems. Also, we compare the results in the standard case, using Belief Function (BF) only and using the proposed method (OBFSoS).

From the results shown in Table 1. The difference between the player's classes can vary due to the presence of different contents. That's according to the

Table 1. # of conflicts generated with Standard Method vs. Belief Function vs. OBFSoS

Simulation attempts	# of conflicts		
	Standard Method	Belief Function	OBFSoS
1	45	40	32
2	43	34	24
3	39	30	25
4	38	30	24
5	35	26	24
6	34	24	17
7	28	20	15
8	25	18	14
9	25	15	10
10	21	12	7

meanings it is the role of semantics. In Fig. 2, the graph shows that the OBFSoS gives a fewer number of conflicts than BF method that also produce fewer conflicts than the standard one.

Table 2. we give a ratio about the conflicts that we can resolve from the results that were in Table 1. There are more conflicts during the 10 attempts varying from the Standard method, BF and OBFSoS one.

In Table 2, we difference conflicts relying on its type. Highlighted conflicts were on decision, design,

Table 2. The ratio of conflicts that resolved with the Standard method vs. the Belief Function method

attempts	Standard Method					Belief Function Method				
	# of conflicts	Ratio				# of conflicts	Ratio			
		decision	design	security	code		decision	design	security	code
1	45	0.311	0.356	0.178	0.156	40	0.300	0.275	0.175	0.250
2	43	0.419	0.326	0.186	0.070	34	0.324	0.265	0.176	0.235
3	39	0.385	0.256	0.128	0.231	30	0.333	0.267	0.133	0.267
4	38	0.316	0.289	0.184	0.211	30	0.267	0.267	0.167	0.300
5	35	0.257	0.229	0.257	0.257	26	0.269	0.269	0.154	0.308
6	34	0.441	0.265	0.176	0.118	24	0.292	0.292	0.125	0.292
7	28	0.286	0.357	0.107	0.250	20	0.300	0.250	0.100	0.350
8	25	0.200	0.240	0.360	0.200	18	0.222	0.222	0.111	0.444
9	25	0.160	0.320	0.280	0.24	15	0.200	0.267	0.133	0.400
10	21	0.286	0.143	0.286	0.286	12	0.167	0.250	0.083	0.500

Table 3. The ratio of conflicts that resolved using the OBFSoS method

attempts	OBFSoS Method						BF Method		
	# of conflicts	Ratio				ratio of conflicts	Resolving time in seconds	ratio of conflicts	Resolving time in seconds
		decision	design	security	code				
1	32	0.267	0.200	0.133	0.111	0.711	220.15	0.889	318.00
2	24	0.233	0.186	0.116	0.0233	0.558	205.23	0.791	270.00
3	25	0.205	0.179	0.103	0.154	0.641	197.26	0.769	247.20
4	24	0.211	0.158	0.132	0.132	0.632	169.01	0.789	234.60
5	24	0.171	0.143	0.200	0.171	0.686	154.36	0.743	228.00
6	17	0.176	0.176	0.088	0.059	0.500	131.00	0.706	210.05
7	15	0.179	0.143	0.036	0.179	0.536	121.09	0.714	199.20
8	14	0.120	0.120	0.160	0.160	0.560	110.07	0.720	186.60
9	10	0.040	0.080	0.120	0.160	0.400	90.47	0.600	162.00
10	7	0.048	0.048	0.190	0.048	0.333	76.80	0.571	156.40

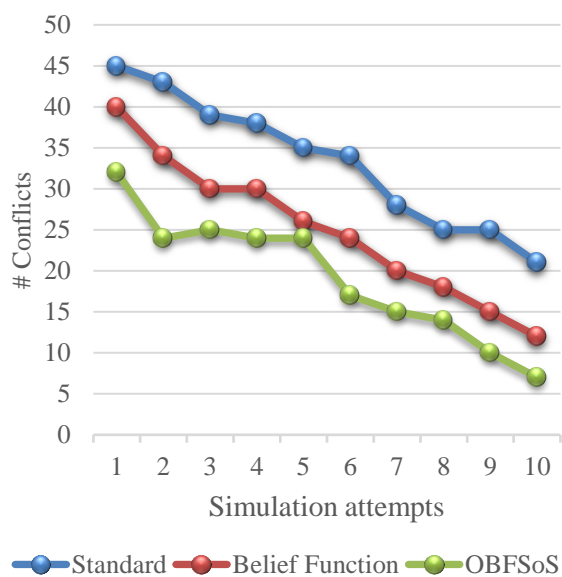


Fig. 2. Standard vs. BF vs. OBFSoS conflicts no. comparison

code and security. These conflicts are common in the SoS we are working on. Conflicts generally resolved using the standard vs. BF ones. The fundamental values that need improvement are the decision in the SoS. The design and code importance is for homogeneous systems to reduce conflicts. Security is sometimes important because our game is an online game.

The standard method in the SoS is the method after integration without improvement as in other methods. In the results in the table, we show that the standard method gives an improvement. This does not mean that it resolves conflicts. But the learning agent found for such smart games we prepared to prevent the similar conflicts during attempts. The BF method implements the proposed Belief algorithm to resolve conflicts but without applying ontology as in OBFSoS.

In Table 3, we present the proposed OBFSoS method compared to the methods in Table 2. We also add the ratio of conflicts for the OBFSoS with BF.

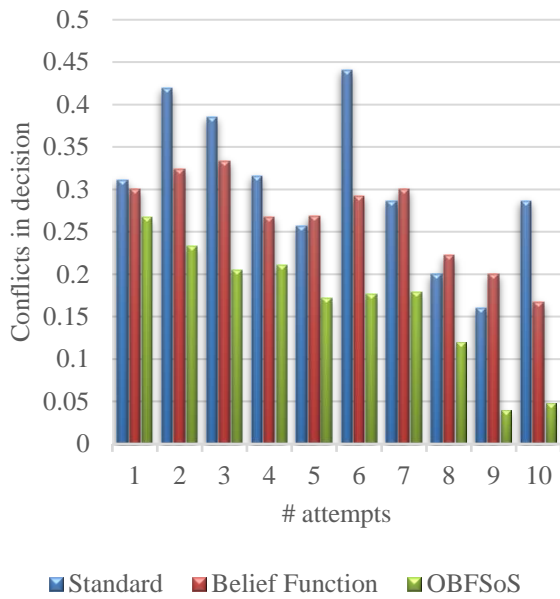


Figure. 3 Standard vs. BF vs. OBFSoS decision conflicts no.

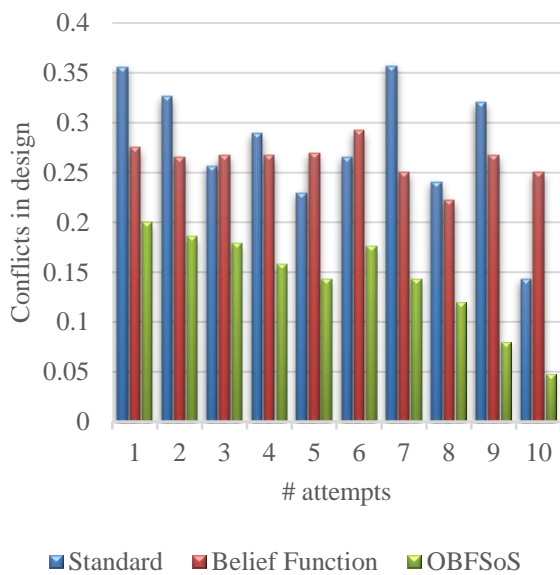


Figure. 4 Standard vs. BF vs. OBFSoS design conflicts no.

Plus, the time of resolving conflicts for the two methods.

Fig. 3 gives a comparison between the ration of the values for conflicts in a decision that classified in the SoS part. The chart as shown clarifies that results when ran SoS, applied the three methods. The BF method reduces conflicts in the decision in a good way without ontology. The OBFSoS tries to reduce conflicts in the decision as shown in the graph better than BF. This is because of ontology usage. It uses dealing with conflicts based on its mechanism.

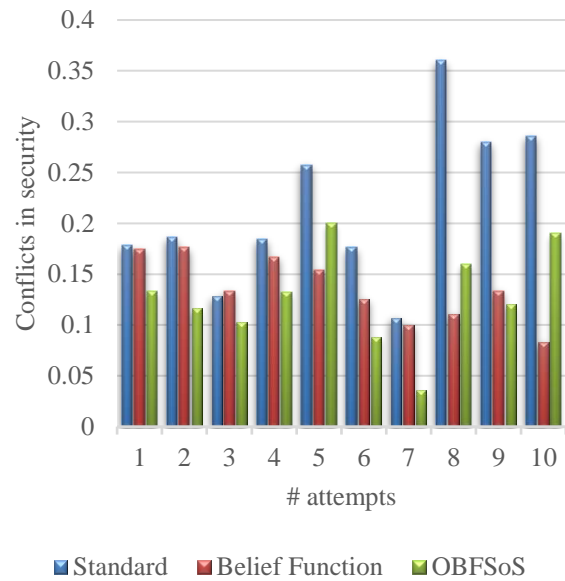


Figure. 5 Standard vs. BF vs. OBFSoS security conflicts no.

In Fig. 4 compares the results of the design conflict. Applied the OBFSoS gives appropriate results compared to the results of other methods.

Decision and design types of conflicts are which we improve the OBFSoS method because they are a part of the decision support system for the SoS. Although that in other types for our application. We try to handle some conflicts of these different types which affects our decision. Our results are close to being better in the cases presented in the charts.

Fig. 5 represents values of the OBFSoS method. This compares values before being applied by the security side to meet conflicts in the SoS part. The ratio varies according to the effect of the SoS decision only. Because the game has an online version. Also, in this case, not all the players can enter the game until passing the registration rules. The case we work for security. When security has effects on the decision, we need to be sure about our information on the score to decide who is the winner. Although OBFSoS have best results but also the BF method is better. This is because Belief function maintains security for its evidence checking which is found also in the OBFSoS.

About the code, we have differences in the results. This depends on reducing conflicts. This depends more on adapting to the game and the integration of the systems using ontology. In implementing the OBFSoS we try to decrease conflicts during solving. Besides that, as we show on Fig. 6, each attempt also gives a better result than the preceding one. This depends on the learning factor in the game. In

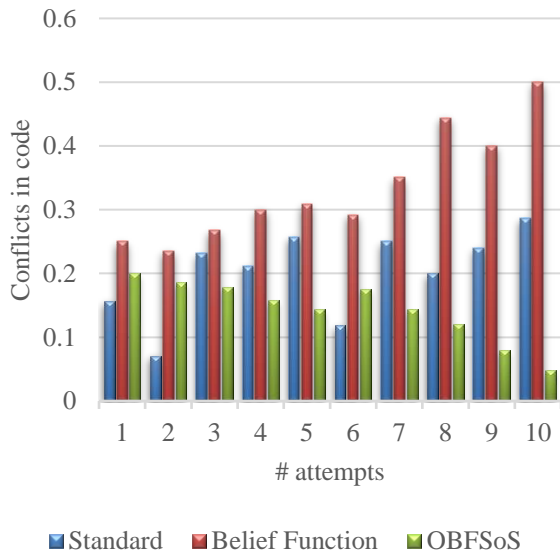


Figure. 6 Standard vs. BF vs. OBFSoS code conflicts no.

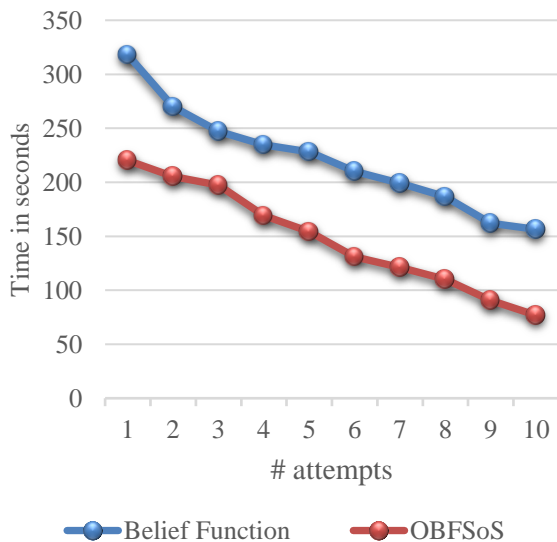


Figure. 7 Conflicts ratio time of resolving for OBFSoS method vs. BF

addition to this learning, the agent avoids some conflicts of other attempts using the OBFSoS. This idea we apply for dealing with conflicts about decision support as resolving decision and design conflicts. Results are better than using BF only without ontology.

Time problem is the main issue in the SoS. When the number of conflicts is less, SoS can be lighter and faster in use. So that we need to save time. As an RTS game, the results of the graph in Fig. 7 show the efficiency of the OBFSoS in conflict detection and rapid handling. We compare the OBFSoS results with the BF one.

Saving time here is useful for such huge organizations as SoS. We give in the chart above to clarify the ratio of conflicts after applying the OBFSoS method. The time measured by seconds that the OBFSoS spent in dealing with conflicts. The chart clarifies that with more attempts OBFSoS can be better at dealing with conflicts than BF. That is based on learning so that the result improves gradually. BF is better but using ontology that is important in some conflict's types. This effects on time for BF.

Finally, we see that the Ontology effects on the conflict's outcomes. It reduces the number of conflicts compared to non-use of the Ontology. The importance of Ontology in decreasing conflicts based on the semantic advantage that belongs to Ontology. It can determine the error is real conflict or just a similarity. For the identification of general conflicts in the SoS. These, such as data conflicts, so that using the ontology can show it. Some conflicts appear with Ontology less than without it because of semantic signs. Other methods can out conflict, but as total don't distinguish each other.

Because the game has an online version. Also, in this case, not all the players have permission to enter the game (Only for security). When security has effects on the decision, we need to be sure about our information on the score to decide who win.

Each attempt also gives a better result than the preceding one. This depends on the learning factor in the game. In addition to this learning, the agent avoids some conflicts of other attempts using the OBFSoS. This idea we apply for dealing with conflicts about decision support as resolving decision and design conflicts.

## 6. Conclusions and future work

The System of Systems perspective for real integration system is important. So, we proposed a method to provide a solution for different types of conflict, which represents an important issue. Because the conflict can destroy SoS ability to give a real right decision in a short time. In our approach, we proposed the OBFSoS method, hybrid Ontology concept with belief function theory after amelioration for SoS. That's to find the optimum solution for the conflict. The proposed method introduces a new equation to calculate the weight. We implemented the method on a case study of an online RTS game as a homogenous SoS. Different types of conflicts appear in homogenous SoS.

The conversion to Ontology has a positive impact on the conflict resolution on the SoS. This step can reduce some semantic conflicts which have not been discovered without its use. Also, using Ontology

doesn't lose information, but keeps it. After executing OBFSoS on RTS game, we solve about 70% of the conflict in the first samples, then we reach about 87%. The last 13% of the conflicts are security, connectivity, and other problems. As a learned-system, making many attempts gives a better result. We taught the agent by attempts during the algorithm. This effect positively on the results. Every time conflict handled; the agent prevents it in the next attempt from the result. It gives it a solution for such conflict. Not only this improves results. Also, it helps in reducing time in discovering conflicts in the SoS.

In the future, we will try to improve the results to solve more conflict-related problems. We will consider implementation in other cases. Also, make the OBFSoS method capable of resolving more conflicts types problems on SoS. In addition to this, we will work on a huge system in SoS such as smart games with big data.

## References

- [1] E. Honour, "Verification and Validation Issues in Systems of Systems", *arXiv preprint arXiv:1311.3626*, pp. 2-7, 2013.
- [2] J. S. Dahmann, G. Rebovich Jr, and J. A. Lane, "Systems engineering for capabilities", *MITRE CORP MCLEAN*, Vol. A, No. 22102-7539, 2008.
- [3] B. Boehm and J. Lane, "Using the incremental commitment model to integrate system acquisition, systems engineering, and software engineering", *CrossTalk*, Vol. 19, pp. 4-9, 2007.
- [4] "System of Systems (SoS)", *PURDUE University*, May2019 Available: <https://engineering.purdue.edu/Engr/Research/Initiatives/Archive/SoS>
- [5] E. Cavalcante, N. Cacho, F. Lopes, T. Batista, and F. Oquendo, "Thinking smart cities as systems-of-systems: A perspective study", In: *Proc. of the 2nd International Workshop on Smart*, pp. 9, 2016.
- [6] V. Dimitriadis, *Reinforcement Learning in Real Time Strategy Games Case Study on the Free Software Game Glest*, Department of Electronic and Computer Engineering Technical University of Crete, China, 2009.
- [7] J. Soyez, G. Morvan, R. Merzouki, and D. Dupont, "Multilevel agent-based modeling of system of systems", *IEEE Systems Journal*, Vol.11, No.4, pp. 2084- 2095, 2015.
- [8] T. Gruber, "A translation approach to portable ontology specifications", *Knowledge acquisition*, Vol. 5, pp. 199-220, 1993.
- [9] I. Mayk and A. Madni, "The role of ontology in system-of-systems acquisition", *Intelligent Systems Technology Santa Monica CA*, 2006.
- [10] H. Yan, Z. Jing, Y. Li-qun, L. Ze-min, and T. Lijian, "Based on ontology methodology to model and evaluate system of systems (SoS)", In: *Proc. of the 9th International Conference on System of Systems Engineering (SOSE)*, pp. 101-106, 2014.
- [11] J. Osmundson, T. Huynh, and P. Shaw, "Developing Ontologies for Interoperability of Systems of Systems", *SOSECE*, 2006.
- [12] G. Abdalla, C. Damasceno, and E. Nakagawa, "A Systematic Literature Review on Systems-of-Systems Knowledge Representation", Technical Report 405 *Department of Computer Science University of São Paulo*, Brazil, 2015.
- [13] Y. Zhang, X. Liu, Z. Wang, and L. Chen, "A Service-Oriented Method for System-of-Systems Requirements Analysis and Architecture Design", *JSW*, Vol. 7, pp. 358-365, 2012.
- [14] A. Dempster, "Upper and lower probabilities induced by a multivalued mapping", *The Annals of Mathematical Statistics*, pp. 325-339, 1967.
- [15] W. Liu, "Analyzing the degree of conflict among belief functions", *Artificial Intelligence*, Vol. 170, pp. 909-924, 2006.
- [16] A. Martin, "About conflict in the theory of belief functions", *Belief Functions: Theory and Applications (ed: Springer)*, pp. 161-168, 2012.
- [17] A. Martin, "Conflict management in information fusion with belief functions", *arXiv preprint arXiv:1709.04182*, 2017.
- [18] A. Pyster, D. Olwell, N. Hutchison, S. Enck, J. Anthony, and D. Henry, "Guide to the systems engineering body of knowledge (SEBoK) v. 1.0. 1", *Guide to the Systems Engineering Body of Knowledge (SEBoK)*, 2012.
- [19] T. de Melo Novaes Viana, "Managing Conflicting Requirements in Systems of Systems", In: *Proc. of the 2017 CRC PhD student Conference*, 2017.
- [20] A. Gorod, M. DiMario, B. Sauser, and J. Boardman, "'Satisficing' system of systems using dynamic and static doctrines", *International Journal of System of Systems Engineering*, Vol. 1, pp. 347-366, 2009.
- [21] J. Lane and R. Valerdi, "Synthesizing SoS concepts for use in cost modeling", *Systems Engineering*, Vol. 10, pp. 297-308, 2007.
- [22] C. Stary, "System-of-systems design thinking on behavior", *Systems*, Vol. 5, p. 3, 2017.
- [23] D. Ormrod, B. Turnbull, and K. O'Sullivan, "System of systems cyber effects simulation

- ontology", In: *Proc. of Winter Simulation Conference (WSC)*, pp. 2475-2486,2015.
- [24] J. Dahmann, G. Rebovich, M. McEvelley, and G. Turner, "Security engineering in a system of systems environment", In: *Proc. of IEEE International Systems Conference (SysCon)*, pp. 364-369,2013.
- [25] K. Baldwin, J. Dahmann, and J. Goodnight, "Systems of Systems and Security: A Defense Perspective", *Insight*, Vol. 14, pp. 11-14, 2011.
- [26] Z. Fang, K. Moolchandani, H. Chao, and D. DeLaurentis, "A Method for Emission Allowances Allocation in Air Transportation Systems from a System-of-Systems Perspective", *Journal of Cleaner Production*, pp. 419-431, 2019.
- [27] C. Langa, "Managing conflict situations within the school system organisation", In: *Proc. of 2018 10th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pp. 1-4, 2018.
- [28] A. Nikas, H. Doukas, and L. M. López, "A group decision making tool for assessing climate policy risks against multiple criteria", *Heliyon*, Vol. 4, pp. e00588, 2018.
- [29] E. Lefevre, O. Colot, and P. Vannoorenberghe, "Belief function combination and conflict management", *Information fusion*, Vol. 3, pp. 149-162, 2002.
- [30] E. Lefèvre and Z. Elouedi, "How to preserve the conflict as an alarm in the combination of belief functions?", *Decision Support Systems*, Vol. 56, pp. 326-333, 2013.