ABC⁺: Extended Action-Benefit-Cost Modeling with Knowledge-based Decision-Making and Interaction Model for System of Systems Simulation

Mingyu Jin
Korea Advanced Institute of Science and Technology
Daejeon, Republic of Korea
mgjin@se.kaist.ac.kr

Donghwan Shin
Korea Advanced Institute of Science and Technology
Daejeon, Republic of Korea
donghwan@se.kaist.ac.kr

Doo-Hwan Bae
Korea Advanced Institute of Science and Technology
Daejeon, Republic of Korea
bae@se.kaist.ac.kr

ABSTRACT

The system of systems (SoS) is a large-scale and complex system composed of autonomous and independent constituent systems (CSs). It deals with complex requirements as SoS-level goals that are not able to be satisfied by a single CS. To analyze SoS-level behaviors considering CSs' autonomy and independence, SoS-level engineers need to model and simulate an SoS properly. Action-Benefit-Cost (ABC) modeling provides an effective and efficient way to model and simulate the autonomous and independent behaviors of CSs by focusing on their external actions, benefits, and costs. However, ABC modeling does not support internal-knowledge of a CS. In addition, ABC modeling does not support communications between CSs. SoS engineers also have difficulty in reflecting all possible interaction into models in advance of simulation. In this paper, we propose ABC⁺ modeling, which supports the three-phased knowledge-based decision-making and the message-based interaction models. The case study shows that ABC⁺ modeling improves the expressiveness of SoS models and lets SoS-level behavior simulation by different interaction patterns be possible, which cannot be even represented in ABC modeling. The simulation execution time is examined about both ABC and ABC⁺ modeling.

CCS CONCEPTS

• **Software and its engineering** → **Software design engineering**;

KEYWORDS

System of Systems; agent-based modeling; Action-Benefit-Cost modeling; decision-making; interaction

ACM Reference format:

Mingyu Jin, Donghwan Shin, and Doo-Hwan Bae. 2018. ABC⁺: Extended Action-Benefit-Cost Modeling with Knowledge-based Decision-Making and Interaction Model for System of Systems Simulation. In *Proceedings of ACM SAC Conference, Pau,France, April 9-13, 2018 (SAC'18),* 4 pages.
https://doi.org/10.1145/3167132.3167426

ACM ISBN 978-1-4503-5191-1/18/04…$15.00
https://doi.org/10.1145/3167132.3167426

1 INTRODUCTION

The increasing scale and complexity of systems have reached a point that imposes qualitatively new demands on the existing system technology. The emerging system is characterized as distributed, decentralized, and networked combinations of autonomous subsystems under large-scale and complex environments. This new “system" is a general concept of “System of systems" (SoS) [6], consisting of autonomous and independent constituent systems (CSs). Complex requirements and goals that are not able to be satisfied by a single system call for the concept of the SoS.

In SoS, CS has unique characteristics such as (1) managerial independence, i.e., each system is managed in most parts for its own goals rather than the goals of SoS, (2) operational independence, i.e., each system operates independently to achieve its own goals by itself [12] and (3) interoperability, i.e., each system is able to exchange information through agreed interfaces, protocols and standards, which helps to achieve its own goals or/and SoS-level goals. To investigate the SoS-level behaviors, it is important to model and simulate the independent CSs properly [14].

To tackle this problem, Kim et al. [9] proposed Action-Benefit-Cost (ABC) modeling which abstracts the complex decision-making mechanisms of independent CSs as a simple utility maximization over observable actions of each CS considering the benefits and costs for the actions. In simulation, each CS performs an action whose utility (e.g., benefit minus cost) is the maximum. The abstraction keeps the autonomy of CSs by hiding the detailed information while reducing the simulation cost dramatically. However, the existing ABC modeling has two limitations: (1) each CS has no knowledge to remember their history and (2) there is no way to interact between CSs.

Suppose a firefighter (i.e., an autonomous CS) as an example, who has several observable actions (such as move up, move down, move left, and move right) to search patients in a close area. When the benefits and costs of the actions are fixed, the firefighter may infinitely revisit the already-visited position in the closed area because of the first limitation. Furthermore, even if each of the firefighters can remember where they visited somehow, there is no way to exchange information about where the other firefighters visited. If there were communication between the firefighters, they can save their time from redundant visits.

In this paper, we propose ABC⁺ modeling, which solves the limitations and completes the existing ABC modeling method. The contributions of this paper are as follows:

- We propose the knowledge-based decision-making and the message-based interaction models to solve the limitations in ABC modeling's expressiveness.
- We provide a case study to see how the simulation results of the ABC⁺ modeling vary from that of the existing ABC modeling in terms of the SoS-level behaviors and the simulation execution time.

In the following, we briefly review four SoS types and their interactions with examples. Section 3 proposes ABC⁺ with supporting models. We provide a case study to see the effectiveness of the method and the effect on the simulation execution time in Section 4. In Section 5, we compare our methodology with other approaches. We conclude the paper with future works in Section 6.

2 BACKGROUND

2.1 System of Systems

Seo et al. [17] organized the characteristics of four SoS types (i.e., *directed*, *acknowledged*, *collaborative*, and *virtual*) which are originally proposed by Maier [12] and later modified by Dahmann et al. [1]. In directed SoS, an SoS-level manager has full authority on CSs, and manages them to achieve SoS-level goals while they still operate independently. For example, when a national disaster occurs, the control tower acts as the SoS manager and manages various resources such as firefighters, ambulances, hospitals, etc. SoS manager in acknowledged SoS request CSs to cooperate by proposing various inducements. This type can be matched a case of middle-level disasters. In collaborative SoS, CSs have agreements on SoS-level goals without a SoS manager. Lastly, a virtual SoS refers to a situation where there is neither SoS manager nor agreed SoS-level goals.

To help readers understand, we bring a simple but intuitive SoS scenario called the Mass Casualty Incident (MCI) response scenario [11]. We suppose a big explosion occurred and buildings are broken down, where people are trapped. Firefighters (i.e., CSs) are dispatched, and each has the ability to search for trapped people and pull out them. Since all firefighters try to pull out as many as possible, the independent firefighters may collaborate. It exemplifies a collaborative SoS. Meanwhile, a government agency may set up a control tower (i.e., SoS manager) for more effective and efficient control. The control tower may make just inducements in an acknowledged SoS or commands in a directed SoS.

3 MODELING WITH ABC⁺

3.1 Knowledge-based Decision-making Model

To solve the limitation that each entity model has no way to remember its history, we attempt to introduce a knowledge base for each entity such as CS or SoS manager. The knowledge base is widely used to simulate the decision-making mechanism of human beings in computer systems [5, 7, 15, 16]. Instead of using the fixed cost-benefit table given by domain experts, ABC⁺ lets entities autonomously build their own independent knowledge, which is used in cost-benefit analysis. Also, we define the three phases of one's decision-making mechanism. The detailed description of them is as follows.

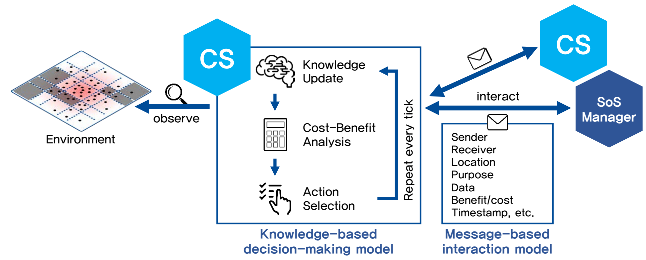

Figure 1: Overview of ABC⁺ Modeling.

Table 1: Purposes of Message

Purpose	Object Type	Description
Request	Information	Request information from an entity
	Action	Request to do an action at an entity
Order	Action	Order an entity to do an action
Respond	Request	Respond to a request from an entity
Deliver	Information	Deliver the information to another entity

Table 2: SoS Types and Interaction Patterns

SoS type	Interaction model
Virtual	No interaction
Collaborative	CS-CS
Acknowledged	CS-CS, CS-SoS manager (weak guidance)
Directed	CS-SoS manager (strong order)

Phase 1: Knowledge update. An entity use available information for decision-making. It is represented as individual and independent knowledge. There can be various kinds of knowledge such as its own action history, observed resource availability or information obtained from other entities. One's knowledge is the set of available information, which has type, value and evaluation functions.

Phase 2: Cost-benefit analysis. In this phase, it calculates benefit & cost of actions with weighted sum of evaluated values of each information in the knowledge. Decisions are rational but bounded onto its own knowledge [4].

Phase 3: Action selection. The entity chooses the best action in the cost-benefit table adjusted in Phase 2. For now, we use the simplest utility function: $utility = benefit - cost$.

3.2 Message-based Interaction Model

To solve the second limitation, we attempt to introduce the interaction model. Interactions between SoS entities are the key source for one entity to get useful information. In ABC⁺ model, every communication is represented in the form of a message. The information is wrapped and delivered with related data such as identification, purpose, etc. To embody the SoS characteristics [10], it includes location, timestamp and trust data in a message. Benefit and cost is as well included to support the intuitive evaluation on some messages like requests.

Every interaction has a purpose–we make requests such as asking something or asking someone to do something, make responses to the requests or simply give information to others. Especially, in a certain situation, someone who has higher authority over another gives commands, which are compulsory, i.e., order. These five categories are listed in Table 1. They have different requirements on response and compulsion.

As explained in Section 2.1, there are four types of SoS: virtual, collaborative, acknowledged, and directed. The communicative relationship between entities is different according to SoS type as shown in Table 2. Collaborative SoS assumes interactions between CSs only while acknowledged and directed SoS consider interactions between CS and SoS manager.

4 EVALUATION

4.1 Research Questions

We consider the following research questions:

- RQ1: What impact do decision-making models have on SoS-level goal achievement?
- RQ2: How does interaction model affect SoS-level goal achievement?
- RQ3: How do ABC and ABC⁺ models affect the simulation execution time?

The RQ1 compares the expressiveness of the existing ABC model and the proposed ABC⁺ model. It focuses on the problematic behavior without knowledge. The RQ2 examines the importance of considering interactions in SoS simulation. More specifically, the ABC⁺ models are expected to be able to express different SoS types with different interaction patterns that may affect SoS goal achievements. The RQ3 addresses the simulation cost of the extended ABC modeling method. We measure how the execution time varies with different SoS types in ABC⁺ modeling.

4.2 Experiment Setting

To evaluate the ABC⁺ modeling, we consider the MCI-Response SoS scenario. The MCI area is represented as the 49x49-sized grid, and each cell represents a building over the area. A big explosion occurred at the center of the area. The explosion damage radiates outward. Total 100 people are trapped inside collapsed buildings. The SoS-level goal is to rescue all the trapped people. Three firefighters are dispatched to deal with the MCI, and each firefighter has the ability to search for people and to pull out them autonomously and independently.

We made five versions; one for ABC model and the others for ABC⁺ differing by different message-based interaction models depending on the SoS type. These are implemented and put into simulation engine in SIMVA-SoS [8, 18], the simulation-based statistical verification & analysis tool. It performs discrete event- and agent-based simulation. To handle non-determinism in models, we performed every experiment 10,000 times, taking average values.

4.3 Analysis Results

4.3.1 RQ1: What impact do decision-making models have on SoS-level behaviors? The simulation results of the ABC modeling may vary depending on the existence of the knowledge bases for each CS. Figure 2 shows that the SoS-level goal achievement does not reach 100% without the knowledge base. On the other hand, ABC⁺ model's goal achievement keeps increasing until 100% pull-out. Note that 100% achievement does not depends on the different SoS models in ABC⁺ modeling as shown in Figure 3. Only adding the knowledge-based model, SoS simulation becomes working properly.

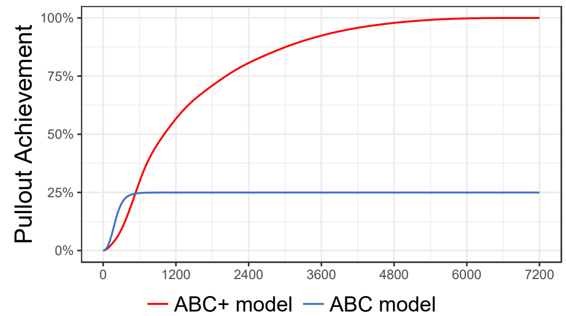

Figure 2: Pull-out goal achievement for ABC and ABC⁺.

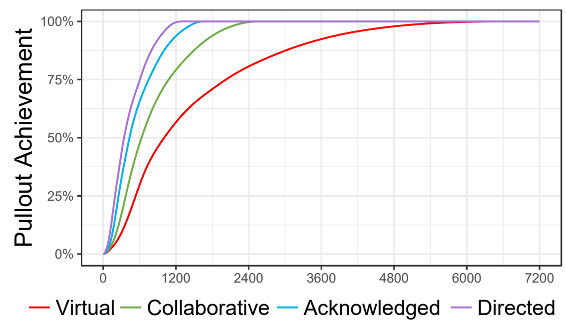

Figure 3: SoS-level goal achievement for different SoS Types.

4.3.2 RQ2: How does interaction model affect SoS-level goal achievement? In addition to the knowledge-based model, we should see that the interaction model is also essential. The message-based interaction model separates the interaction behaviors from one's decision-making model, and makes modeling different interactions easier than before. With ABC⁺ modeling, we made four interactions models considering the four SoS types.

Figure 3 shows the different trends of SoS-level goal achievement. While all types of SoS achieves 100% of the goal at the end of the simulation, the very moment when each SoS achieves the goal is different; virtual SoS takes 7129 ticks, collaborative SoS takes 2752 ticks, acknowledged SoS takes 1737 ticks and directed SoS takes 1383 ticks. This simulation results confirm that the different message-based interaction model affects in the SoS-level goal achievement as expected.

4.3.3 RQ3: How do ABC and ABC⁺ models affect the simulation execution time? Figure 4 shows the average execution time for each SoS type. Interestingly, the execution time of virtual and acknowledged SoS outnumbered that of collaborative and directed SoS. It means that the existence of SoS manager greatly influences the execution time. This is because every tick during the simulation,

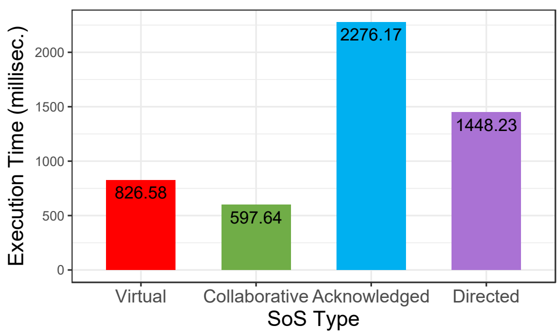

Figure 4: Simulation execution time for different SoS types.

the control tower requests&receives pull-out knowledge to/from all firefighters, and it sends guidance or messages to each of them.

Another interesting point lies between virtual and collaborative. Considering that a collaborative SoS additionally cater for the interactions between CSs compared to a virtual SoS, the fact that the execution time of a virtual takes longer than that of a collaborative SoS is interesting. This is because the knowledge sharing between CSs reduces the redundant actions of CSs.

5 RELATED WORK

MAPE-k feedback loop is a popular architecture for the autonomic computing with the knowledge base models proposed by IBM [5]. It is used to build the structure of self-managing systems such as self-healing or self-optimizing. It requires detailed information to derive a proper decision-making model. However, ABC⁺ models abstract the detailed internal decision-making by focusing on the external behaviors with the corresponding benefits and costs.

Multi-Agent Systems (MAS) are distributed systems composed of entities called agents, similar to SoSs. Agent communication languages (ACLs), such as FIPA ACL [3] and knowledge query and manipulation language (KQML) [2], are the de-facto ACLs for communication between knowledge-based systems or agents. They define the concept of the message and protocol. They also describe a message by their own BNF grammar rules. Their focus is to provide standards for actual implementation of MAS. ABC⁺ modeling omits some categories like negotiation and error handling, which are realized by performatives of FIPA ACL. However, we can safely omit some details for easiness but SoS-specific details needed for successful approximation. In addition, ACLs do not successfully consider SoS-specific details suggested by Kopetz [10] such as physicalism, neutrality, temporal aspects.

6 CONCLUSIONS

As various domains have demands on the large-scale and complex systems such as SoS, modeling and simulation methods which represent target systems in a proper level of approximation are required for system engineers. ABC modeling method is proposed with the economic-rational view, considering the lack of accessibility to detailed decision-making mechanism. However, it has limitations on modeling realistic decision-making process and interactions between entities.

In this paper, we proposed ABC⁺ modeling with the knowledge-based decision-making process and the message-based interaction model. The case study with the MCI-Response example shows that the proposed model has a proper model for autonomous and interoperable decision-making and is capable of modeling different types of SoS and interaction patterns and simulating different level of SoS-level goal achievements. It also shows how the simulation execution time varies with the interaction patterns.

The knowledge-based decision-making and the message-based interaction model has various directions for further researches. Game theory can be used in the analysis phase to calculate benefits and costs considering uncertainty [13]. In addition, it is worth to study what messages should contain and how messages should be interpreted for better and realistic simulations.

ACKNOWLEDGMENTS

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the SW Starlab support program(IITP-2017-R0126-17-1101) supervised by the IITP(Institute for Information & communications Technology Promotion).

REFERENCES

[1] Judith S Dahmann and Kristen J Baldwin. 2008. Understanding the current state of US defense systems of systems and the implications for systems engineering. In *Systems Conference, 2008 2nd Annual IEEE*. IEEE, 1–7.
[2] Tim Finin, Richard Fritzson, Don McKay, and Robin McEntire. 1994. KQML as an agent communication language. In *Proceedings of the third international conference on Information and knowledge management*. ACM, 456–463.
[3] FIPA Fipa. 1997. *specification part 2: Agent communication language*. Technical Report. Technical report, FIPA-Foundation for Intelligent Physical Agents.
[4] Nigel Gilbert. 2008. Agent-Based Models, Quantitative Applications in the Social Science153. *London, Thousand Oaks* (2008).
[5] Paul Horn. 2001. Autonomic computing: IBM\'s Perspective on the State of Information Technology. (2001).
[6] Mohammad Jamshidi. 2011. *System of Systems Engineering: Innovations for the Twenty-First Century*. John Wiley & Sons.
[7] Jeffrey O Kephart and David M Chess. 2003. The vision of autonomic computing. *Computer* 36, 1 (2003), 41–50.
[8] Junho Kim. 2017. *Model-based System of Systems Verification considering Selfishness*. Master's thesis. Korea Advanced Institute of Science and Technology, Daejeon, Korea.
[9] Junho Kim, Donghwan Shin, and Doo Hwan Bae. 2017. An Applicability Study of Action-Benefit-Cost Model and Statistical Model Checking for System of Systems Goal Achievement Verification. *KIISE Transactions on Computing Practices* 23 (2017), 256–261.
[10] Hermann Kopetz. 2014. A conceptual model for the information transfer in systems-of-systems. In *Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC), 2014 IEEE 17th International Symposium on*. IEEE, 17–24.
[11] David MacGarty and David M Nott. 2014. *Disaster Medicine: A Case Based Approach*. Springer Science & Business Media.
[12] Mark W Maier. 1996. Architecting principles for systems-of-systems. In *INCOSE International Symposium*, Vol. 6. Wiley Online Library, 565–573.
[13] Edward J Mishan and Euston Quah. 2007. *Cost-benefit analysis*. Routledge.
[14] Claus Ballegaard Nielsen, Peter Gorm Larsen, John Fitzgerald, Jim Woodcock, and Jan Peleska. 2015. Systems of Systems Engineering: Basic Concepts, Model-Based Techniques, and Research Directions. *ACM Computing Surveys (CSUR)* 48, 2 (Sept. 2015), 1–41.
[15] Anand S Rao and Michael P Georgeff. 1991. Modeling rational agents within a BDI-architecture. *KR* 91 (1991), 473–484.
[16] Anand S Rao, Michael P Georgeff, et al. 1995. BDI Agents: From Theory to Practice.. In *ICMAS*, Vol. 95. 312–319.
[17] Dongwon Seo, Donghwan Shin, Young-Min Baek, Jiyoung Song, Wonkyung Yun, Junho Kim, Eunkyoung Jee, and Doo-Hwan Bae. 2016. Modeling and verification for different types of system of systems using PRISM. In *Software Engineering for Systems-of-Systems (SESoS), 2016 IEEE/ACM 4th International Workshop on*. IEEE, 12–18.
[18] SESoS. 2017. SIMVA-SoS. https://github.com/SESoS/SIMVA-SoS. (2017).