Data-Driven Environment Modeling for Adaptive System-of-Systems

Yong-Jun Shin, Young-Min Baek, Eunkyoung Jee, and Doo-Hwan Bae
Korea Advanced Institute of Science and Technology (KAIST)
Daejeon, Republic of Korea
{yjshin,ymbaek,ekjee,bae}@se.kaist.ac.kr

ABSTRACT

Since a System-of-Systems (SoS) is constructed and managed under a complex and dynamic environment, self-adaptability has become one of the key capabilities that SoSs should have. To design an adaptive SoS, analyzing and modeling the environment are important. Studies on self-adaptive systems (SAS) have proposed various analysis and design approaches to deal with dynamic environment and operating conditions. However, most existing approaches require a considerable amount of domain experts' knowledge about the operating environment without specific and practical guidelines, so there still remain many challenges for engineers to analyze and design an adaptive SoS. In this study, we propose a data-driven method of generating environment models for adaptive SoS. To guide the analysis and understanding of the environment, we propose a metamodel that encompasses characteristics of the dynamic environment. Based on the metamodel, an environment model is generated from historical data for effective analysis of the SoS's complex environment. As a case study, we apply our method to a traffic environment modeling with real data. We show that our proposed method can practically help engineers generate environment models with concrete features that are necessary for adaptive SoS modeling by considering the environment as a major entity for SAS analysis and design.

CCS CONCEPTS

• **Software and its engineering** → **Software design engineering**; • **Computing methodologies** → **Modeling methodologies**;

KEYWORDS

self-adaptive system, adaptive System-of-Systems, environment modeling, data-driven analysis, smart traffic system, modeling methodology

ACM Reference Format:
Yong-Jun Shin, Young-Min Baek, Eunkyoung Jee, and Doo-Hwan Bae. 2019. Data-Driven Environment Modeling for Adaptive System-of-Systems. In *The 34th ACM/SIGAPP Symposium on Applied Computing (SAC '19), April 8–12, 2019, Limassol, Cyprus.* ACM, New York, NY, USA, Article 4, 4 pages. https://doi.org/10.1145/3297280.3297618

SAC '19, April 8–12, 2019, Limassol, Cyprus

ACM ISBN 978-1-4503-5933-7/19/04.
https://doi.org/10.1145/3297280.3297618

1 INTRODUCTION

A System-of-Systems (SoS) is a large complex system consisting of multiple constituent systems to achieve goals that cannot be solved by a monolithic system [11]. As the goals and requirements that modern systems have to satisfy become more complicated, the complexity of the environments interacting with SoSs also increases. Therefore, SoS should be designed and developed as a system that can adapt its behavior to the dynamic and complex environment, so-called a self-adaptive system [13]. To design an adaptive SoS, analyzing and modeling SoS environment are important, because the environment interacts closely with an SoS, like a physical environment of a mass casualty incident (MCI) response SoS [2].

Various analysis and design approaches for SASs have been proposed and most of them require engineers to have sufficient domain knowledge of the system's environment [1, 3, 4, 7, 8]. However, it is difficult to follow the approaches when the engineers' knowledge is not sufficient or the environment is too complex like an adaptive SoS's environment, because there is a lack of practical guidelines of analyzing the environment. As the environment that the system should adapt to becomes more complicated with a number of dynamic and uncertain environmental factors, practical analysis and design approaches are necessary. Also, the approaches should consider analyzing the environment as one of the major steps in the modeling process.

In this study, we propose a data-driven method for adaptive SoS environment modeling. To guide the analyzing and understanding of the environment, we propose a metamodel that encompasses the characteristics of the SAS environment. Based on the metamodel, environment models can be generated using historical environment data following our modeling process. This method utilizes environment data to identify concrete features of the complex environment in designing adaptive SoS. As a case study, we apply our method to a traffic environment modeling with real data.

2 RELATED WORK

Although there is a lack of literature describing the relationship between SoS and its environment, Baek, *et al.* briefly explained the environment as an interacting component with SoS [2]. In terms of adaptation, there have been many efforts to define the characteristics of the environment for adaptive system modeling. One main characteristic of the environment is that the environment dynamically changes its states or behaviors, and the SAS adapts its behaviors to the dynamic changes in environment. Many modeling approaches have tried to design the SAS analyzing the dynamic changes of the environment [1, 3, 4, 7, 12]. Another characteristic is that the changes in the environment affect the achievement of system goals. Many SAS design methods regard managing or

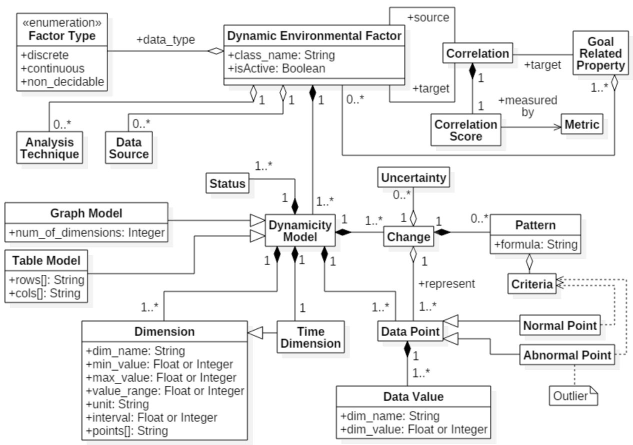

Figure 1: A metamodel of environment for adaptive SoSs

minimizing the effects on the goals caused by the environment as a purpose of SAS design [5–8].

Based on the characteristics of the environment, analyzing and understanding the environment are required prior to the SAS design. Morandini, *et al.* explicitly described the environment modeling method, but the environment model is not for a general understanding of the environment [8]. In addition, even though many modeling approaches require sufficient analysis of the environment, they only partially provide practical guidelines to analyze the environment or generate the environment model [1, 3, 9]. Therefore, it is hard to apply the existing modeling approaches to the complex and huge environment of SoS. In this study, we show how to model the complex environment to provide sufficient understanding of environmental characteristics for adaptive SoS modeling based on a practical data analysis method.

3 DATA-DRIVEN ENVIRONMENT MODEL GENERATION FOR ADAPTIVE SOS

3.1 Metamodel for data-driven environment model of adaptive SoS

We propose a data-driven environment model generation method for adaptive SoSs. Our method provides guidelines to SoS engineers to generate environment models using historical data. This method includes practical modeling processes to analyze and understand the environment, assuming that engineers do not have sufficient domain knowledge of the environment and they lack capabilities for data analysis. First, We propose a metamodel of the environment so that engineers can generate environment models that fully reflect the characteristics of the SAS environment, by providing environmental components, related entities, and relationships among them.

The environment consists of a number of environmental factors. We focuses on environmental dynamicity, and Figure 1 shows a metamodel of the *dynamic environmental factor*s that change their status or behaviors. A *dynamicity model* is a major component of the dynamic environmental factor that represents information about changes in the environmental factor. Based on the environmental data, it has multiple *dimension*s of the data like a time dimension. Because *time* is almost always included in the environmental data, it is treated as an essential dimension in our metamodel. The environmental *data point*s represent the *change*s of the environmental factor and the changes may have a *pattern* with specific *criteria*. If the environmental factor shows a patterned change, each data point can be categorized as a *normal point*, or an *abnormal point*. In addition, the dynamicity of the environmental factor can be represented as a *graph* or a *table* to help engineers visually analyze the situation according to the analysis style.

A dynamic environmental factor may have a *correlation* with other factors, and an explicit expression of this relationship implies internal mechanisms of the environment within the range of known historical data. In addition, a factor can have a correlation to *goal-related properties* to express an effect of the environmental factor on the adaptive SoS's goal.

The metamodel shows features of the adaptive SoS environment. The information on historical data, dimensions of data, changes in environmental factors, the correlation between environmental factors, and relationships between the environment and the goal-related properties are shown in the metamodel. Some of these features can be directly determined by engineers, but the others might have to be analyzed by data analysis using real data. In the next section, we explain how an environment model is generated using historical data based on the metamodel introduced.

3.2 Environment model generation process

Figure 2 shows the environment model generation process. The first step of environment model generation is acquiring historical environment data which are related to the target adaptive system. Thanks to the trends to share the data publicly, we can easily access to massive environmental data. The input historical data of specific environmental factors should be given. The environmental data, which are organized as a set of environmental factors' data, become the basis of the environment model.

The second step is to identify factor types and analysis techniques of the environment model. The factor type is determined by the format of the data and an analysis technique is determined depending on the factor type. The analysis technique of a factor can be statistical methods or algorithms that will be used for data analysis. Following this step, the data should be preprocessed in machine-readable form.

In the fourth step, Some factors might have to be marked or processed by engineers if there are goal-related factors of interest in the acquired and preprocessed data. Because one of the goals of adaptive SoS is to adapt or control the system in the changing environment, the results of the adaptation affect the environment interactively [6, 10]. Therefore, some of the environmental factors can indirectly give insights about the relationship between an adaptive SoS and its environment. For example of a traffic control system design, a goal to reduce traffic jam can be reflected in the average speed of cars. The average speed of cars can be marked as a goal-related factor of the traffic environment.

From the fifth through seventh steps are the data analysis process that generates information in the dynamicity model component for each environmental factor. As the first step of the data analysis,

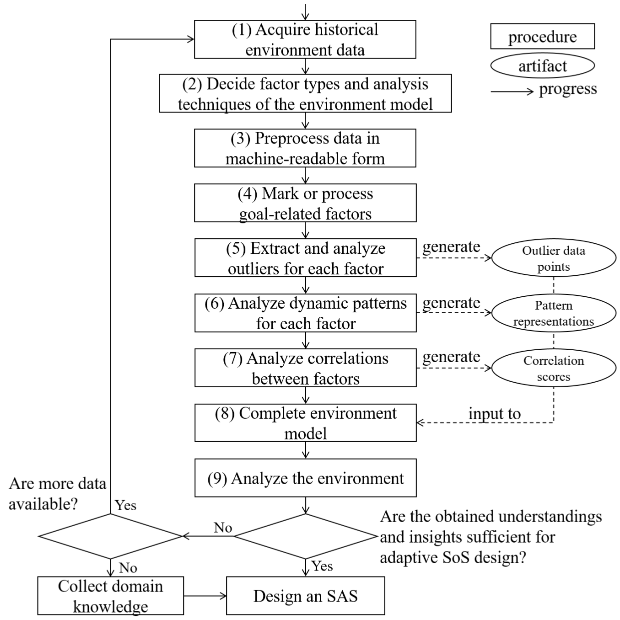

Figure 2: An environment model generation process

outliers of the data are extracted and analyzed. Even if the change has a dominant pattern, there might exist some outliers that do not match the pattern. Those abnormal data of the environment should be extracted and be carefully considered in adaptive SoS modeling to handle unexpected conditions. A specific threshold or an outlier classification metric such as a Z-score[1] can be used to identify an outlier.

In the next step, the changes in each environmental factor and the patterns should be analyzed. By regarding a data point as a status of the environmental factor, the changes of the data can be analyzed as a pattern if they have a distinct dominant flow of changes. The pattern is expressed by a polynomial equation or state-based model if the status of the factor is represented as a continuous value or a discrete value, respectively. Polynomial regression can be used to generate a pattern of continuous changes as an equation. State-based model, such as the Markov-chain model, can be used to express discrete changes in environmental factors.

After analyzing the changes in environmental factors, analyzing correlations of the environmental factors is conducted in step 7. Analyzing relationships with goal-related factors is also included in this step. Typically, the correlation between two factors can be analyzed using the Pearson correlation coefficient. The correlation score of two factors have value between -1 and +1, where (+) indicates positive correlation, and (-) indicates negative correlation.

After the data analysis process, the analyzed artifacts, which are outliers, patterns, and relationships of environmental factors, are combined and complete the environment model based on the metamodel. These artifacts are model components defined in the metamodel. The data analysis process and model completion process can be automated.

Based on the generated environment model, an engineer can get basic understandings and insights of the environment. If the generated environment model is sufficient to analyze the environment, system design can be performed accordingly with the concrete features in the environment model. Following the environment model generation process, an engineer can utilize the environment data to efficiently analyze the dynamic and complex environment for adaptive SoS design.

4 CASE STUDY: REAL TRAFFIC ENVIRONMENT MODELING

In order to show the applicability of the proposed method to SoS environment, we show a real traffic environment modeling in the case study. For a realistic scenario, we adopt a specific target environment of Daejeon, South Korea. We acquired the historical traffic data of the target area from the Daejeon Transportation Data Warehouse system (DTDW 2.0), which is managed by Daejeon city[2]. The acquired dataset, which contains 18 dynamic factors including the average speed, the number of cars, and the average occupancy of the streets over time, was used to generate the environment model. We generated the traffic environment model of the given historical data of the target area following our model generation method. A part of the generated environment model including only three factors is shown in Figure 3.

This environment model can provide engineers with specific information about the features of the traffic environment including outliers, dynamic pattern, and correlations between factors. Authors without expert knowledge on traffic were able to understand the situation of the target area by following the data analysis process of the environment model generation. Application of this traffic environment model to a smart traffic control system design from requirement analysis was conducted. The result was uploaded on our website[3].

5 CONCLUSION

An adaptive SoS modeling approach considering the environment as a first-class entity is necessary to attain a deep understanding of the environment during the design of an adaptive SoS. We proposed a data-driven environment modeling method for adaptive SoS, to practically analyze its very complex environment. The proposed method is based on the metamodel that provides concrete features to be used for adaptive SoS analysis and design. The environment model is generated using historical data of the environment based on the metamodel and the data analysis guidelines. In the case study, we showed that our method can guide engineers to generate an environment model. We have shown that our approach can practically help engineers to obtain the essential understanding of the environment required for the adaptive SoS analysis and design with data analysis methods.

[1] $z = \frac{x-\mu}{\sigma}$, where μ is the mean of the population, and σ is the standard deviation of the population. A data point whose absolute value of z-score is larger than a threshold is regarded as an outlier.

[2] DTDW 2.0 - http://tportal.daejeon.go.kr

[3] http://se.kaist.ac.kr/starlab/studies/study-1-sos-and-environment-modeling/2-environment-modeling-of-sos/

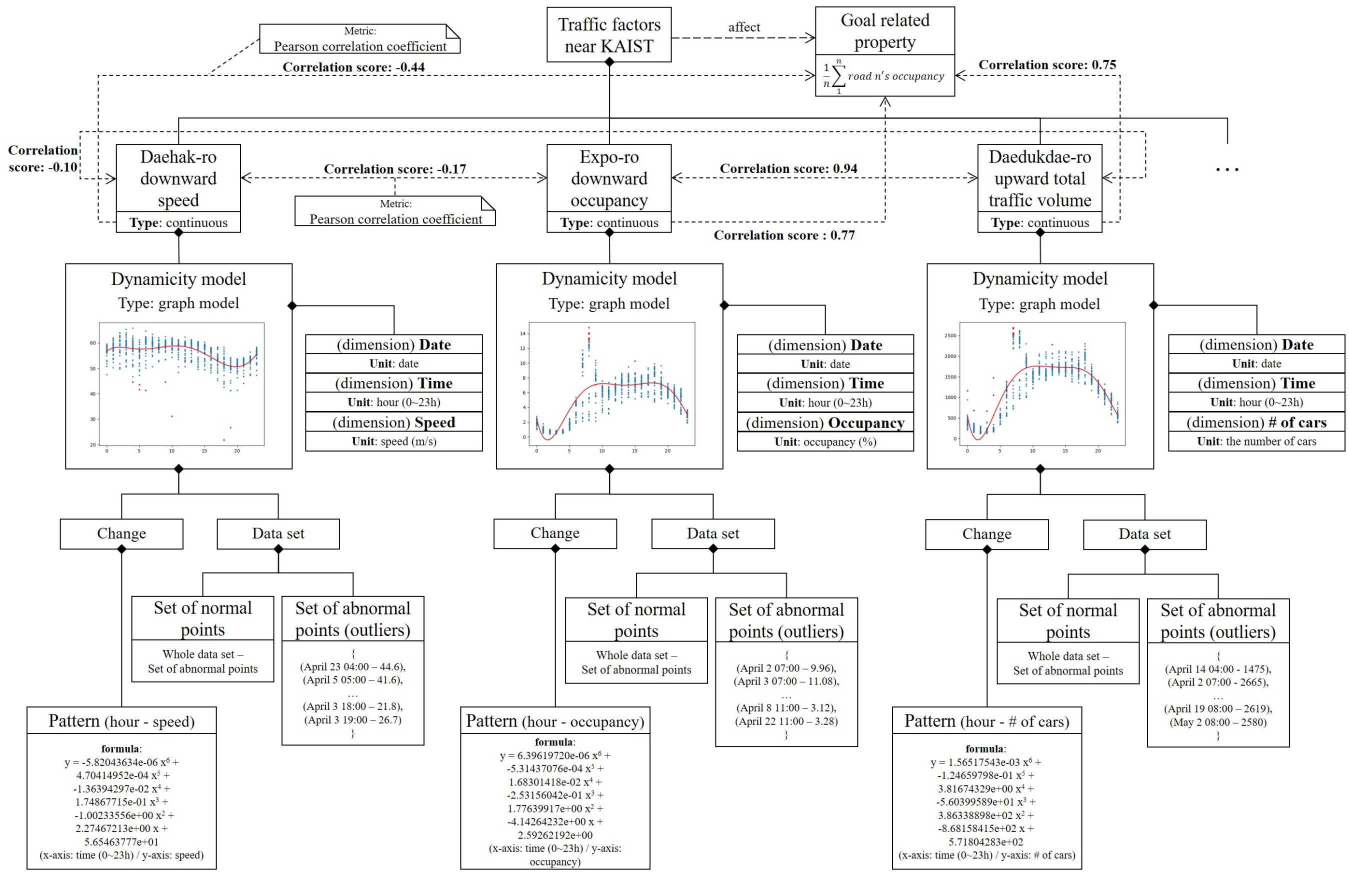

Figure 3: A traffic environment model generated using the given historical traffic data

ACKNOWLEDGEMENTS

This research was supported by the Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No. 2015-0-00250, (SW Star Lab) Software R&D for Model-based Analysis and Verification of Higher-order Large Complex System), and Next-Generation Information Computing Development Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT (2017M3C4A7066212).

REFERENCES

[1] Paolo Arcaini, Elvinia Riccobene, and Patrizia Scandurra. 2015. Modeling and analyzing MAPE-K feedback loops for self-adaptation. In *Proceedings of the 10th international symposium on software engineering for adaptive and self-managing systems*. IEEE Press, 13–23.

[2] Young-Min Baek, Jiyoung Song, Yong-Jun Shin, Sumin Park, and Doo-Hwan Bae. 2018. A meta-model for representing system-of-systems ontologies. In *2018 IEEE/ACM 6th International Workshop on Software Engineering for Systems-of-Systems (SESoS)*. IEEE, 1–7.

[3] Betty HC Cheng, Pete Sawyer, Nelly Bencomo, and Jon Whittle. 2009. A goal-based modeling approach to develop requirements of an adaptive system with environmental uncertainty. In *International Conference on Model Driven Engineering Languages and Systems*. Springer, 468–483.

[4] Zuohua Ding, Yuan Zhou, and Mengchu Zhou. 2018. Modeling Self-Adaptive Software Systems by Fuzzy Rules and Petri Nets. *IEEE Transactions on Fuzzy Systems* 26, 2 (2018), 967–984.

[5] Ahmed Elkhodary, Naeem Esfahani, and Sam Malek. 2010. FUSION: a framework for engineering self-tuning self-adaptive software systems. In *Proceedings of the eighteenth ACM SIGSOFT international symposium on Foundations of software engineering*. ACM, 7–16.

[6] Antonio Filieri, Martina Maggio, Konstantinos Angelopoulos, Nicolás D'Ippolito, Ilias Gerostathopoulos, Andreas Berndt Hempel, Henry Hoffmann, Pooyan Jamshidi, Evangelia Kalyvianaki, Cristian Klein, et al. 2015. Software engineering meets control theory. In *Proceedings of the 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*. IEEE Press, 71–82.

[7] Mirko Morandini, Loris Penserini, and Anna Perini. 2008. Towards goal-oriented development of self-adaptive systems. In *Proceedings of the international workshop on Software engineering for adaptive and self-managing systems*. ACM, 9–16.

[8] Mirko Morandini, Loris Penserini, Anna Perini, and Alessandro Marchetto. 2017. Engineering requirements for adaptive systems. *Requirements Engineering* 22, 1 (2017), 77–103.

[9] Tharindu Patikirikorala, Alan Colman, Jun Han, and Liuping Wang. 2012. A systematic survey on the design of self-adaptive software systems using control engineering approaches. In *Software Engineering for Adaptive and Self-Managing Systems (SEAMS), ICSE Workshop on*. IEEE, 33–42.

[10] Andres J Ramirez and Betty HC Cheng. 2009. Evolving models at run time to address functional and non-functional adaptation requirements. In *Proceedings of the 4th International Workshop on Models at Runtime*.

[11] Andrew P Sage. 2011. *System of systems engineering: innovations for the 21st century*. Vol. 58. John Wiley & Sons.

[12] Moeka Tanabe, Kenji Tei, Yoshiaki Fukazawa, and Shinichi Honiden. 2017. Learning environment model at runtime for self-adaptive systems. In *Proceedings of the Symposium on Applied Computing*. ACM, 1198–1204.

[13] Danny Weyns. 2017. Software engineering of self-adaptive systems: an organised tour and future challenges. *Chapter in Handbook of Software Engineering* (2017).