DESIGNING EMERGENCE IN SYSTEMS OF SYSTEMS USING INFORMATION STREAMS

Hochmann, Uriel;
Reich, Yoram

Tel Aviv University

ABSTRACT

Systems complexity is increasing, in particular, when systems become systems of systems (SoS). SoS are composed of constituent systems (CS), have unique goals for the CS and the SoS as a whole, and present new SoS-level properties called emergent properties. Emergent properties are unique because they only appear at the system level. Current research has only revealed some tools focusing on simple emergence for engineers aiming to design emergence at the SoS level. However, forming design tools for the creation or modification of strong emergence will enable engineers to create systematic changes in the SoS. This article proposes a connection between emergence and information streams with the latter being a model of the transfer of information between the different CS in the SoS. A methodology for designing SoS information streams is demonstrated, with encouraging results, using a multi-agent simulation of the propagation of the COVID-19 virus through citizens. By testing several information stream configurations, an SoS with a decrease of 47% in sick agents was found. These results show that by changing the information stream better SoS performance is attained, supporting designing in a complex world.

Keywords: Systems Engineering (SE), Simulation, Decision making, Systems of Systems, Emergent Properties

Contact:
Hochmann, Uriel
Tel Aviv University
Israel
yurielnam@gmail.com

Cite this article: Hochmann, U., Reich, Y. (2023) 'Designing Emergence in Systems of Systems Using Information Streams', in *Proceedings of the International Conference on Engineering Design (ICED23),* Bordeaux, France, 24-28 July 2023. DOI:10.1017/pds.2023.136

1 INTRODUCTION

A System of Systems (SoS) is a collection of constituent systems (CS) integrated to accomplish one or multiple shared goals. An SoS is defined when the components are independent in several characteristics (Rainey & Jamshidi, 2018): operational independence, developmental independence, and geographic distribution. The study of emergence within SoS investigates ways to create system-level phenomena that contribute value that is not attainable by a single CS. This research will focus on the existing gap in tools for creating emergence in SoS through designing the information transferred between CS in the SoS, also referred to in this research as an information stream.
This paper aims to provide a theory of positive emergence that can guide the development of tools to engineer positive emergent properties or behaviors. These tools will enable SoS Engineers (SoSE) to design desired system-wide properties through high-order interaction of CS. Resilience, adaptability, and SoS services are some selected emergent properties. The theory will relate to the connection between information stream manipulation and positive emergence.
The structure of this paper is as follows; the problem space is defined in the background section including a literature review to show some existing tools for emergence design. Section 3 explains the specific goals for our tool of emergence design given the gaps highlighted in the review. Section 4 explains the methodology for this research. Section 5 shows the results of a simulation showcasing the proposed tool and Section 6 presents the conclusions arising from the research and points to some possible future directions.

2 BACKGROUND

SoS are central in the modern era, from smart cities to defense systems, providing unique services only attainable through intricate systems. SoS have components that are developed and operated by separate businesses to achieve specific goals. One such example is an airport; the airplane is a single system designed to carry passengers from one terminal to another. Other systems, such as baggage ticketing systems, handle the baggage identifier, so each is routed to a specific plane. By working together, these systems achieve the shared goal of passenger and cargo transport from one location to the next.
There is a growing interest in manipulating SoS to produce wanted results. However, current system engineering tools do not effectively address SoS. Engineering emergent behaviors that only manifest at the SoS level due to the interaction of the CS, remain a critical challenge for SoSE (Dahmann, 2014; Valerdi et al., 2008). While there have been multiple works for framework, architecture and modeling, and simulation, there has been very little research on a design methodology of emergence in SoS. Specifically, the review conducted for this research found only a single approach (Watson et al., 2020).
This research aims to construct a theory for developing a tool for engineering emergent behaviors. The proposal presents an integrated review of tools and techniques developed for engineering emergent properties and behaviors. Building upon existing methods, information stream modeling is introduced. Information stream analysis allows the identification of underused information in an SoS. Increasing the utility of underused information in the SoS enables the system to optimize and introduce novel emergent properties. A use case analysis will show how this technique encourages positive emergence.
Engineering emergent properties allows the creation of high-order effects in SoS. In the context of modeling and simulation, emergent properties have four levels of complexity (Tolk et al., 2018) : *Simple emergence*: Simplified system models enable the prediction of simple emergent properties. The system subcomponents are clear and well-understood. Each system segment can be isolated and reproduced. An example of a positive reaction is the information transfer rate within a computer network. An adverse reaction is the resulting emergent smell arising from molecule interactions in chemical compounds. These can be separated and analyzed by the simulated interaction using a few molecule models.
Weak emergence: the emergent property is reproducible and consistently predicted with simulation. These properties cannot be reduced to a few components within the system but are still available for modeling and simulation. An example of a positive reaction could be the life quality of citizens living within a city. A model containing the entire set of components in the system is required to simulate this quality. It is still possible to replicate in a simulation but not as readily as a simple emergent property. An adverse reaction could be the air quality of citizens living in the city. As the city grows, more components contribute to pollution with interactions between nearby systems contributing to pollution, such as the adjacency of the water system to the electricity plant producing power for the city.

https://doi.org/10.1017/pds.2023.136 Published online by Cambridge University Press

Strong emergence: the emergent property is consistent with known properties but not reproducible in simulations. It is unpredictable and inconsistent in simulation. The economic boom allows conceptualizing strong emergence. There is no way it can be simulated correctly. In hindsight, it is possible to analyze occurrences of economic boom and find the components contributing to that event. An economic contraction could be an example of a negative emergent property due to different CS failing in a specific way that caused the entire financial ecosystem to shrink.
Spooky emergence: The emergent property is inconsistent with the known properties of the system. The interaction between the different components is not understood. There is no clear link between the interaction and the emergent property. Due to this complexity, it is not easy to understand if the emergent property is positive or negative. An example of spooky emergence is human cognition or human culture. Both examples brought here have been positive and negative in past occurrences.

Table 1 - Article analysis. X marks if the article addresses the topic in the column. The analysis shows a shift from framework to architecture and design methodology with consistent modeling, simulation, and case studies. In most articles, authors used case studies to illustrate the theory practically

#	*Journal*	*Article*	*Framework*	*Architecture*	*Design Methodology*	*Modeling and Simulation*	*Case Study*	*Emergence Type*
1	JMD	(Watson et al., 2020)			X		X	Weak
2	Sensors	(Bemthuis et al., 2020)		X				Weak
3	IEEE Access	(Kerr et al., 2020)		X		X		General
4	JSS	(Hachem et al., 2020)		X		X	X	Weak/Strong
5	SE	(Oquendo, 2019)		X				Simple
6	ACM CPS	(Ceccarelli et al., 2019)			X		X	Weak
7	JSEP	(Mori et al., 2018)				X	X	Weak
8	SE	(Sitton & Reich, 2018)	X				X	Simple/Weak
9	JDMS	(Zeigler, 2016)		X			X	Weak
10	IEEE SJ	(Qiu et al., 2014)				X	X	Weak
11	JDMS	(Helle et al., 2013)	X			X		Simple/Weak
12	SAE Tech	(McMurran & Jones, 2013)	X				X	Weak
13	IEEE SJ	(Tsilipanos et al., 2013)				X		Weak/Strong
14	IEEE TSMC	(Rokkas et al., 2012)				X	X	Simple
15	IEEE SJ	(Haghnevis & Askin, 2012)	X					Simple
16	IJSoSE	(McKay et al., 2011)				X	X	Simple/Weak
17	JIE	(DeLaurentis & Ayyalasomayajula, 2009)				X	X	Weak

To lay the groundwork, we conducted a literature review using Scopus and Web of Science, focusing on guided emergence. Searching for articles regarding emergence is complex due to the widespread use of the word in different domains unrelated to SoSE. Searching emergence on Scopus yields 322,401 results for the title, abstract or key. The search found 35 articles that include "system of systems" or "systems of systems" in the title, abstract or key, and "emergence" or "emergent" in the keywords. Adding "guided emergence" in the title, abstract, or key added six articles. Of the 41 results, 17 were irrelevant (not in English or unrelated to emergent properties). The remaining 24 articles were thoroughly read; six had no relation to emergent properties, and one duplicate was removed. Table 1 notes the different contributions of a framework, architecture, design methodology, and modeling and simulation of the 17 articles. An additional column notes if a case study is present in the article.

Framework-related references

Previous work created frameworks that enable the creation of system architecture for SoS. Simple or weak properties can be designed or modified by these frameworks with a bottom-up approach. One example is the modeling of business processes in enterprise SoS through the different elements that take

https://doi.org/10.1017/pds.2023.136 Published online by Cambridge University Press

part in the process. These architectures are then validated using simulation with a model-based approach (Haghnevis & Askin, 2012; Helle et al., 2013; McMurran & Jones, 2013; Sitton & Reich, 2018)

Architecture-related references

Research investigating the effect of system architecture on emergence yielded some insights. Work has been conducted to show how to design weak emergent properties into system architecture, as in the previous section using a bottom-up approach (Bemthuis et al., 2020; Kerr et al., 2020). Other research investigated how an architecture could promote positive emergence using information flow or system constraints (Johnson et al., 2013; Oquendo, 2019; Zeigler, 2016).

Design methodology-related references

Watson et al. (2020) suggested mimicking existing biological SoS to change the design of an engineered SoS; the SoS Engineer can recreate in an artificial SoS an emergent property that exists in the natural system. This was shown by adding a recycling CS to an existing ecosystem.
SoSE can analyze different emergent properties through evolution scenarios of the SoS and design components to enhance these emergent properties. Using a case study of a power grid, Ceccarelli et al., (2019) analyzed the different evolution scenarios, located communal threats, and enabled the design of countermeasures to mitigate the danger, focusing on the weak emergent property of shutdowns.

Modeling and simulation-related references

There have been several attempts to leverage modeling and simulation (M&S) tools for emergence engineering in SoS. Some work has shown the possibility of M&S for sensitivity tests. Works conducted in sensitivity testing usually revolve around measuring specific emergent properties in a bottom-up method and changing CS to increase effectiveness (DeLaurentis & Ayyalasomayajula, 2009; McKay et al., 2011; Rokkas et al., 2012) .
Other articles inspected failures and risks at the SoS level including cascading failures or multiple systems interrupting with other CS within the SoS (Hachem et al., 2020; Tsilipanos et al., 2013) . The last type of work conducted in M&S focuses on system scenario simulation. In these simulations, a specific system scenario is modeled and tested (Mori et al., 2018; Qiu et al., 2014).

Literature review discussion

For the different levels of emergence, four levels of engineering activities are tested: detecting, recreating, enhancing existing emergence, and designing new emergence. Detection is the ability to measure an emergent property at the SoS level. Recreation of emergence refers to the ability to simulate emergence. Enhance existing is the ability to make modifications to existing emergence. Designing new emergence enables the introduction of specific emergence into an existing SoS. The gap in SoSE is shown in Table 2. Most of the articles in the literature review discussed simple and weak emergence as these types of emergence are directly related to the CS level interactions.

Table 2 - Knowledge gap in emergence engineering for SoS by activity.

<table>
<tr><th></th><th>Detect</th><th>Recreate (Simulate)</th><th>Enhance existing</th><th>Design new</th></tr>
<tr><td>Simple</td><td colspan="4" rowspan="2">Simple and weak emergence have tools for detection, recreation, enhancement, and design due to direct interactions (Szabo & Teo, 2015)</td></tr>
<tr><td>Weak</td></tr>
<tr><td>Strong</td><td colspan="2">It is possible to detect and recreate strong emergence by simulation (Maier, 2014; McKay et al., 2011)</td><td colspan="2">Knowledge Gap - No Design tools for strong emergence</td></tr>
<tr><td>Spooky</td><td>Improbable (Maier, 2014)</td><td colspan="3">Not possible by definition</td></tr>
</table>

Most of the articles focused on modeling and simulation showing ways of system modeling to evaluate specific emergence. Only two articles specifically aimed at presenting a design methodology for enhancing SoS emergence. The literature review pointed out a possible definition of factors to induce emergence. According to (Johnson et al., 2013) there are several factors contributing to the induction of emergence in SoS. These factors include the rate of information received from the

https://doi.org/10.1017/pds.2023.136 Published online by Cambridge University Press

environment, reception rate, and degrees of freedom. The rate of information received, and reception rate are the basis of information stream formalism. The information stream is an emergent property of itself and is a result of the system's parameters.

Knowledge gap

Our literature research shows only a few design methodologies for engineering simple and weak emergence in SoS. In addition, there are no specific rules to enable SoS emergence (Valerdi et al., 2008). Similarly, there are no tools to measure or control factors for emergence, and no empirical study shows the connection between emergence and information (Johnson et al., 2013).

3 DESIGNING EMERGENCE

While the tools presented in the literature aim to enhance existing simple and weak emergence in SoS, current SoSE research lacks specific tools to enhance or design novel strong emergence in an SoS. Creating a framework or design method for direct strong emergence engineering will unlock new possibilities in the field of SoSE.

This research aims to demonstrate how a specific emergent property can be created or encouraged through modification of the SoS information streams. In theory, modeling the SoS information streams can serve as a basis for benchmarking. Later, the modification of these information streams will enable the introduction and modification of emergent properties. If such a theory for information streams in SoS exists, it could create new engineering possibilities for emergent properties. Future work will focus on developing a theory that ties information streams and emergence in SoS including a causal relation between information streams. This relation can be tested through case studies and simulations of SoS.

Based on this possible theory, we aim to define a framework to propose new information streams for an existing SoS of interest. This new framework will utilize the theory to introduce or modify emergence in SoS. In future research, we aim to introduce the information stream as an extension of a modeling language and test its efficacy through a use case that evaluates the enhanced language and showcases its usability.

We expect to show a relationship between information stream configuration and emergence in an SoS. SoSE will be able to test the information streams in their SoS to detect discrepancies and implement corrective actions. While this article does not explore the full engineering process, it is a possible usage of the proposed framework. This article will demonstrate the capability to simulate an SoS using ABM (Agent-Based Modeling) with the addition of emergence into an existing system using information stream manipulation.

4 METHODOLOGY

To enable SoSE to use information streams for SoS design, a process must be created to support this design activity. This process aims to form the controls for strong emergence through information stream manipulation. Because SoS are complex, the process will use quantitative data from simulations to conclude about qualitative system-level phenomena.

The starting point of this process is the existence of a system model. The model must contain the relevant CS functionality and at least a preliminary definition of existing information and the information currently transferred between the different CS within the SoS. During the design effort different design alternatives for information streams can be tested through exploratory analysis. However, due to the time required to change information streams in an SoS, it is impractical to test the proposed changes on an actual system in the scope of this research.

The methodology presented includes 5 steps (presented in Figure 1): Initially, the strong emergent property is selected (1). Based on the emergent property, the information stream is defined with relevant metrics (2). The structure and metrics of the information stream are analyzed and a theory about the contribution of the information stream to emergence within the scope of the SoS is created and subsequently refined (3). A tool for the control of the information stream in the SoS is formed with defined opportunities for modification (4). A comparison between the SoS and the related information stream models is conducted and a tool for controlling the chosen emergence for that specific SoS is defined (5).

https://doi.org/10.1017/pds.2023.136 Published online by Cambridge University Press

The system model with a relevant information stream model can be simulated using any programming language. In the example presented in this article, we have used the Simpy library of the Python programming language due to its simplicity and existing Agent-Based simulation capabilities.

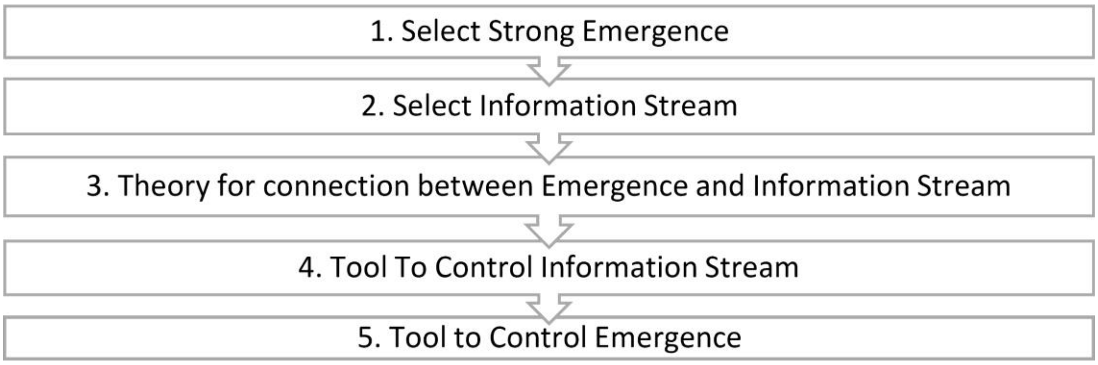


Figure 1 – The research process.

To test the approach, a simple system was selected. The system is a model of a community coping with a sickness such as the COVID-19 virus. The selected strong emergence is resilience, a strong emergent property due to its top-down effect on the system (1 in Figure 1). A less resilient system stops interaction faster due to the sick agents being removed from the interaction. The different citizens in the model travel and generate work hours given that certain conditions are met. When the system is more susceptible to infection, the agent's ability to move throughout the system is reduced. Thus, a top-level property affects the ability of a CS to operate within the context of the system.
The transition of data between the different agents was defined as the information stream (2), and the connection between the effectiveness of the system and the information stream was shown (3). In this case, the tool to control the information stream was simply new connections between agent property and other agents (4) and thus, this tool enables the indirect control of the emergent property (5).

5 RESULTS

The following simulation showcases how engineers can use information stream engineering to affect emergent properties. The text below describes the simulation following the structure presented in the ODD (Overview, Design concepts, Details) protocol (Grimm et al., 2006) to increase readability.
The model's basic idea is a simple simulation of disease propagation through a society. The overall purpose of the model is to show how information flowing through an ABM can affect the emergent property, in this case, the group's resilience to sickness; specifically, pinpointing changes required to the information stream in the SoS to decrease disease propagation in the community. The model includes only entities of type citizen. Citizen entities have six variables: workplace location, home location, cumulative sick time, whether the citizen is showing symptoms, whether the citizen is in quarantine, or if the citizen is ill (see Table 3).

Table 3 - State variables characterizing the citizen entity.

Variable	*Description*	*Possible Values*	*Units*
Workplace	Location of workplace	Set of coordinates	-
Home	Location of home	Set of coordinates	-
Time sick	Duration a citizen is sick; used to determine recovery time	0 – max t	Hours
Symptomatic?	Indicates whether a citizen is showing symptoms	True/False	-
Quarantine?	Indicates whether a citizen is in quarantine	True/False	-
Sick?	Indicates whether a citizen is sick	True/False	-

Every step in the model represents one hour of activity, with a maximum run time of 500 hours, setting the temporal resolution. The simulation landscape is 50X50 unit squares, where multiple citizens could share the same unit square. The population size is 100 citizens, out of which two citizens are sick. The most critical process of the model, which the simulation repeats on every step, is checking for each citizen its logical test to decide if it must enter quarantine. The process then moves the citizens between their homes and workplaces and infects citizens adjacent to other sick citizens if

https://doi.org/10.1017/pds.2023.136 Published online by Cambridge University Press

the citizens are not quarantined. The most important design concept of the model is the connection between a decision and an information source within the information stream modeling.

The statistical distribution of citizens in households and workers in workplaces from (Shi et al., 2010) was used to calibrate the simulation for several reasons. Firstly, the CDC (Centers for Disease Control and Prevention) included it as part of the references for a community mitigation guideline (Qualls et al., 2017) which increase credibility. Additionally, it is presented in a replicable and comparable manner that allows replication of our simulation based on the referenced document. Specifically, we allocated individual citizens' homes and workplaces based on these distributions. This approach allowed us to simulate the spatial distribution of the population in the modeled system and test its effects on the outcomes of interest.

In this model, infection between citizens is set to have a chance at every time step if the citizens share a common space. To simplify the simulation, an infection can only occur when the agents share the same location (home or workplace) and not during transit (which was assumed to happen in less than an hour and so neglected in this model).

Different simulation runs had different information stream configurations (Figure 2). During the exploration phase the simulation was run for all combinations of the information stream and out of all these combinations the following three were selected to demonstrate the connection between information streams and emergence.

In the 'basic' configuration, citizens had information about their proximity to other sick citizens. They began quarantine if they were adjacent to ill citizens (Figure 2(a)).

In the second configuration, 'home 10', citizens had information about proximity to other sick citizens as in the 'basic' configuration and the total citizens sharing a single home (Figure 2(b)). The decision rule was that if a citizen shares a home with 10 or more other citizens, he is quarantined in addition to the 'basic' rule already in place. These two logic rules are evaluated separately for each agent in subsequent order with a logical OR between them.

In the third configuration, 'workplace 20', citizens had information about proximity to other sick citizens as in the 'basic' configuration and the total number of citizens sharing a single workplace. The decision rule was that if a citizen shares a workplace with 20 or more other citizens, he is quarantined in addition to the 'basic' rule already in place. It is important to note that in 'workplace 20' and 'home 10' citizens are kept quarantined even if they had not encountered a sick agent at all.

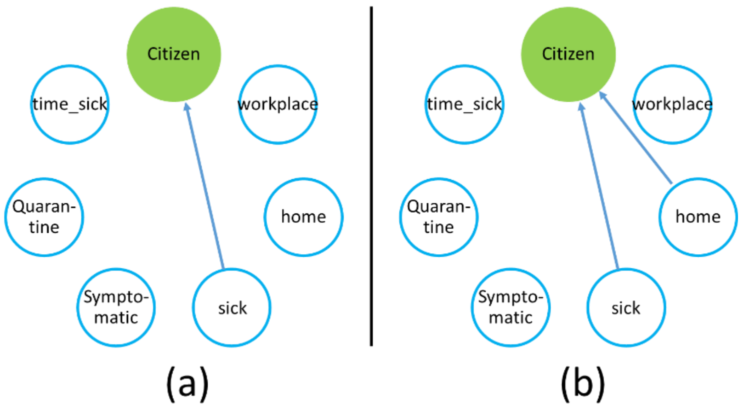
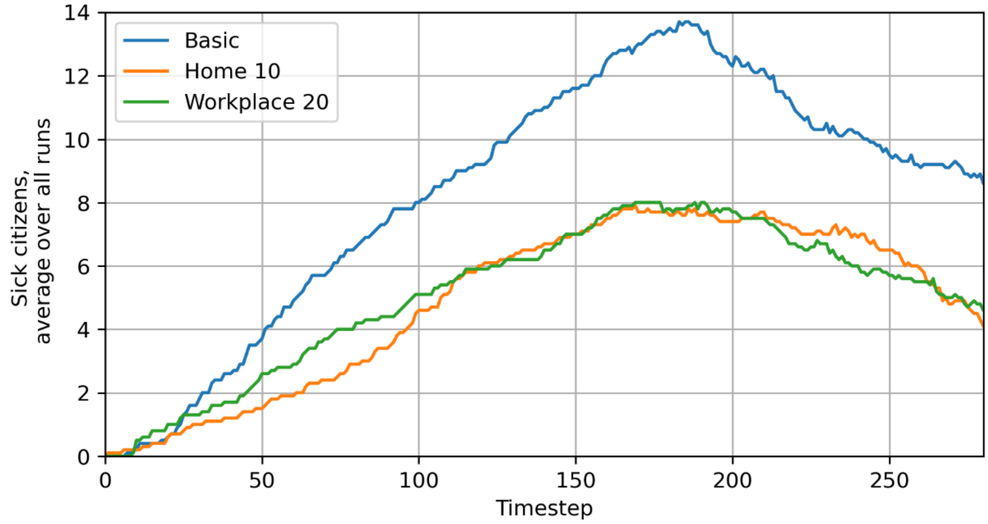


Figure 2 - Two different information stream configurations. (a) citizens only have basic information about the sickness status of other citizens. (b) every citizen also has information about the home locations of other citizens.

From a system design standpoint, the goal is to keep working hours at a maximum while reducing sick citizens to a minimum. These goals contradict, as the ideal solution for the reduction of sick citizens is to keep every citizen at home for the duration of the simulation (lockdown). However, in this ideal solution, no work hours are generated at all. In contrast, the three aforementioned information stream configurations present trade-offs between sick people and the work hours generated. Each configuration was run 10 times with random starting conditions. An average was calculated for the relevant metrics (see Figure 3): sick count and accumulated work hours. We do not present statistical tests as we run each configuration only 10 times.

https://doi.org/10.1017/pds.2023.136 Published online by Cambridge University Press

The baseline information stream configuration, where agents only went to quarantine if exposed to nearby sick agents, denoted by 'basic', has an average that reaches 14 sick citizens. Both suggested information stream alternatives reduce this number to 8.
However, the total work hours for 'Workplace 20' are higher (9200 total) than the 'basic' configuration (8800 total) due to spreading infection while the total work hours for the 'Home 10' are lower than the original configuration (8200 total) due to unnecessarily repeated quarantine of agents. This comparison shows that for maximum work hours, agents should only consider the original information stream and that in high clusters of agents, it is beneficial to automatically quarantine all involved.



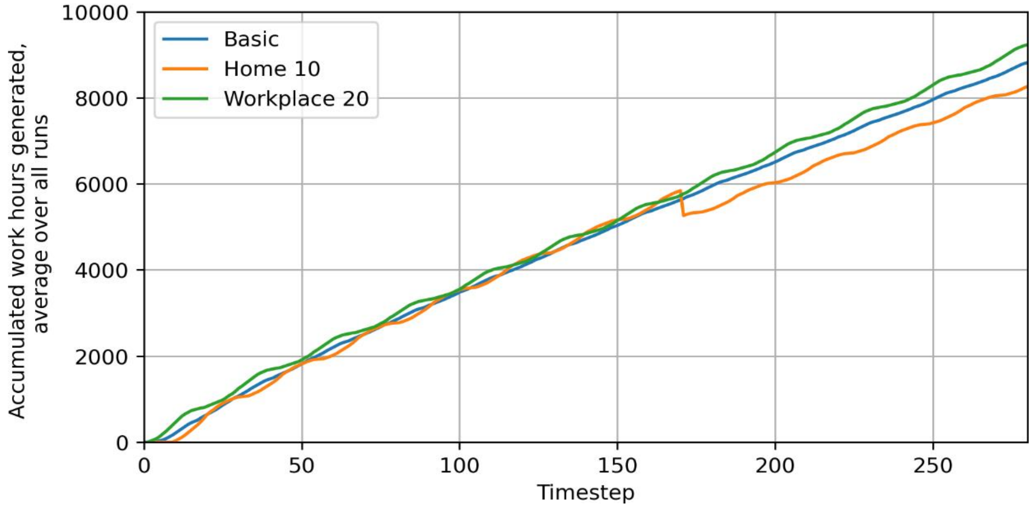


Figure 3 - Simulation results (showing the average for 10 independent runs for each condition); The upper graph shows the average number of sick citizens for every condition while the lower graph shows the accumulated work hours for each condition. Both extra conditions yield a more resilient system than the unmodified version ('Basic') while checking homes ('Home 10') reduces accumulated work hours versus checking workplaces ('Workplace 20')

6 CONCLUSIONS

This article attempts to provide a method to engineer emergent properties into systems of systems. Some previous research presented the use of simulation to generate simple and weak emergent properties but there was no research into strong emergent properties. While there has been some work to show the effect of information stream on emergent properties, there have been no simulations to demonstrate this connection. Such a method will enable system-level engineered properties using a top-down approach. If such a method exists, SoSE can select this property and test the related information exchange in the system through simulation. According to the results of the simulation, different changes can be made to the information stream in the system to achieve the wanted property.

We demonstrated this capability on a simple system. This system was composed of agents that can affect each other but have different goals. The information stream in the demonstrated system was tested; changes in the information stream alone resulted in a decrease of up to 43% in affected agents showing a significant change in the system dynamics due to this information change.

The results of the simulation demonstrated the usage of the proposed method for engineering emergence. The authors suggest that this method can be expanded to other systems of systems as a method of emergent property engineering. This proposed method may be used to model and analyze any system using a coarse approach as the first base for modeling while adding detail as the simulation effort progresses to achieve more accurate results. Future research can expand this work to model specific emergent properties and the required information stream to support them.

As an example of future use, suppose an intelligence SoS is composed of sensor arrays, data analysis systems, and additional components. The different CS have different goals and objectives (e.g., the sensors aim to collect data while the analysis systems aim to analyze and produce actionable information), these CS may extend resources from the global resource pool to increase their throughput (or release resources back to the global resource pool to reduce their throughput). The information stream of this proposed system can be analyzed to enhance self-organization in the SoS as a strong emergent property. A preliminary demonstration of enhancing self-organization is presented in (Hochmann & Reich, 2022). Other information exchanges within the system can be analysed to test their connection to the proposed emergent property or other emergent properties. This enables a simple perspective of the system that enables SoS design using simulation.

BIBLIOGRAPHY

Bemthuis, R., Iacob, M.-E., & Havinga, P. (2020). A Design of the Resilient Enterprise: A Reference Architecture for Emergent Behaviors Control. *Sensors*, *20*(22), 6672. https://doi.org/10.3390/s20226672

Ceccarelli, A., Zoppi, T., Vasenev, A., Mori, M., Ionita, D., Montoya, L., & Bondavalli, A. (2019). Threat Analysis in Systems-of-Systems. *ACM Transactions on Cyber-Physical Systems*, *3*(2), 1–24. https://doi.org/10.1145/3234513

Dahmann, J. (2014). 1.4.3 System of Systems Pain Points. *INCOSE International Symposium*, *24*(1), 108–121. https://doi.org/10.1002/j.2334-5837.2014.tb03138.x

DeLaurentis, D. A., & Ayyalasomayajula, S. (2009). Exploring the Synergy Between Industrial Ecology and System of Systems to Understand Complexity. *Journal of Industrial Ecology*, *13*(2), 247–263. https://doi.org/10.1111/j.1530-9290.2009.00121.x

Grimm, V., Berger, U., Bastiansen, F., Eliassen, S., Ginot, V., Giske, J., Goss-Custard, J., Grand, T., Heinz, S. K., Huse, G., Huth, A., Jepsen, J. U., Jørgensen, C., Mooij, W. M., Müller, B., Pe'er, G., Piou, C., Railsback, S. F., Robbins, A. M., … DeAngelis, D. L. (2006). A standard protocol for describing individual-based and agent-based models. *Ecological Modelling*, *198*(1–2), 115–126. https://doi.org/10.1016/j.ecolmodel.2006.04.023

Hachem, J. el, Chiprianov, V., Babar, M. A., Khalil, T. al, & Aniorte, P. (2020). Modeling, analyzing and predicting security cascading attacks in smart buildings systems-of-systems. *Journal of Systems and Software*, *162*, 110484. https://doi.org/10.1016/j.jss.2019.110484

Haghnevis, M., & Askin, R. G. (2012). A Modeling Framework for Engineered Complex Adaptive Systems. *IEEE Systems Journal*, *6*(3), 520–530. https://doi.org/10.1109/JSYST.2012.2190696

Helle, P., Giblett, I., & Levier, P. (2013). An integrated executable architecture framework for System of Systems development. *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, *10*(4), 435–448. https://doi.org/10.1177/1548512913477259

Hochmann, U., & Reich, Y. (2022). Leveraging Simulation to Improve Self-Organization in Intelligence, Surveillance, and Reconnaissance Systems of Systems. *Proceedings of GAZIT Institution for Intelligence, Strategy, and Technology*, 94–104.

Johnson, J. J., Tolk, A., & Sousa-Poza, A. (2013). A Theory of Emergence and Entropy in Systems of Systems. *Procedia Computer Science*, *20*, 283–289. https://doi.org/10.1016/j.procs.2013.09.274

Kerr, C., Jaradat, R., & Ibne Hossain, N. U. (2020). Battlefield Mapping by an Unmanned Aerial Vehicle Swarm: Applied Systems Engineering Processes and Architectural Considerations From System of Systems. *IEEE Access*, *8*, 20892–20903. https://doi.org/10.1109/ACCESS.2020.2968348

Maier, M. W. (2014). The Role of Modeling and Simulation in System of Systems Development. In *Modeling and Simulation Support for System of Systems Engineering Applications* (pp. 11–41). Wiley. https://doi.org/10.1002/9781118501757.ch2

McKay, S., Mahulkar, V., & Adams, D. (2011). A process to comprehend critical system-of-systems factors: applied to wireless technology design on a navy ship. *International Journal of System of Systems Engineering*, *2*(4), 257. https://doi.org/10.1504/IJSSE.2011.043860

McMurran, R., & Jones, R. P. (2013, April 8). *Robustness Modelling of Complex Systems - Application to the Initialisation of a Hybrid Electric Vehicle Propulsion System*. https://doi.org/10.4271/2013-01-1231

Mori, M., Ceccarelli, A., Lollini, P., Frömel, B., Brancati, F., & Bondavalli, A. (2018). Systems-of-systems modeling using a comprehensive viewpoint-based SysML profile. *Journal of Software: Evolution and Process*, *30*(3), e1878. https://doi.org/10.1002/smr.1878

Oquendo, F. (2019). Architecting exogenous software-intensive systems-of-systems on the internet-of-vehicles with SosADL. *Systems Engineering*, *22*(6), 502–518. https://doi.org/10.1002/sys.21521

Qiu, S., Sallak, M., Schon, W., & Cherfi-Boulanger, Z. (2014). Modeling of ERTMS Level 2 as an SoS and Evaluation of its Dependability Parameters Using Statecharts. *IEEE Systems Journal*, *8*(4), 1169–1181. https://doi.org/10.1109/JSYST.2013.2297751

Qualls, N., Levitt, A., Kanade, N., Wright-Jegede, N., Dopson, S., Biggerstaff, M., Reed, C., Uzicanin, A., Levitt, A., Dopson, S., Frank, M., Holloway, R., Koonin, L., Rasmussen, S., Redd, S., de la Motte Hurst, C., Kanade, N., Qualls, N., Rainey, J., … Reed, C. (2017). Community Mitigation Guidelines to Prevent Pandemic Influenza — United States, 2017. *MMWR. Recommendations and Reports*, *66*(1), 1–34. https://doi.org/10.15585/mmwr.rr6601a1

Rainey, L. B., & Jamshidi, M. (Eds.). (2018). *Engineering Emergence*. CRC Press. https://doi.org/10.1201/9781138046412

Rokkas, T., Neokosmidis, I., Katsianis, D., & Varoutas, D. (2012). Cost Analysis of WDM and TDM Fiber-to-the-Home (FTTH) Networks: A System-of-Systems Approach. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *42*(6), 1842–1853. https://doi.org/10.1109/TSMCC.2012.2227999

Shi, P., Keskinocak, P., Swann, J. L., & Lee, B. Y. (2010). The impact of mass gatherings and holiday traveling on the course of an influenza pandemic: a computational model. *BMC Public Health*, *10*(1), 778. https://doi.org/10.1186/1471-2458-10-778

Sitton, M., & Reich, Y. (2018). EPIC framework for enterprise processes integrative collaboration. *Systems Engineering*, *21*(1), 30–46. https://doi.org/10.1002/sys.21417

Szabo, C., & Teo, Y. M. (2015). Formalization of Weak Emergence in Multiagent Systems. *ACM Transactions on Modeling and Computer Simulation*, *26*(1), 1–25. https://doi.org/10.1145/2815502

Tolk, A., Diallo, S., & Mittal, S. (2018). Complex systems engineering and the challenge of emergence. In *Emergent Behavior in Complex Systems Engineering* (pp. 78–97). John Wiley & Sons, Inc. https://doi.org/10.1002/9781119378952.ch5

Tsilipanos, K., Neokosmidis, I., & Varoutas, D. (2013). A System of Systems Framework for the Reliability Assessment of Telecommunications Networks. *IEEE Systems Journal*, *7*(1), 114–124. https://doi.org/10.1109/JSYST.2012.2207274

Valerdi, R., Axelband, E., Baehren, T., Boehm, B., Dorenbos, D., Jackson, S., Madni, A., Nadler, G., Robitaille, P., & Settles, S. (2008). A research agenda for systems of systems architecting. *International Journal of System of Systems Engineering*, *1*(1/2), 171. https://doi.org/10.1504/IJSSE.2008.018137

Watson, B. C., Malone, S., Weissburg, M., & Bras, B. (2020). Adding a Detrital Actor to Increase System of System Resilience: A Case Study Test of a Biologically Inspired Design Heuristic to Guide Sociotechnical Network Evolution. *Journal of Mechanical Design*, *142*(12). https://doi.org/10.1115/1.4048579

Zeigler, B. P. (2016). A note on promoting positive emergence and managing negative emergence in systems of systems. *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, *13*(1), 133–136. https://doi.org/10.1177/1548512915620580