Time Granularity in System-of-Systems Simulation of Infrastructure Networks

Conference Paper

Author(s):
Dubaniowski, Mateusz Iwo; Heinimann, Hans Rudolf

Publication date:
2018

Permanent link:
https://doi.org/10.3929/ethz-b-000419668

Originally published in:
Studies in Computational Intelligence 812, https://doi.org/10.1007/978-3-030-05411-3_39

Time granularity in system-of-systems simulation of infrastructure networks

Mateusz Iwo Dubaniowski[1][0000-0003-3601-7296] and Hans R. Heinimann[1][0000-0002-8808-2682]

[1] ETH Zurich, Future Resilient Systems at the Singapore-ETH Centre (SEC), which was established collaboratively between ETH Zurich and National Research Foundation (NRF) Singapore (FI 370074011), under the auspices of the NRF's Campus for Research Excellence and Technological Enterprise (CREATE) programme.
Singapore-ETH Centre, Future Resilient Systems, 1 CREATE Way, #06-01 CREATE Tower, Singapore 138602, Singapore.
`iwo.dubaniowski@frs.ethz.ch`

Abstract. Because of their extreme complexities, a system-of-systems (SoS) approach is often used for simulating infrastructure systems. This allows the user to integrate models of various systems into one simulation. However, this integration presents several issues because individual simulations are often designed for only a specific purpose and context. This leads to variations among space granularities and proposes a challenge when selecting an appropriate time granularity for the overall SoS simulation.

To explore how this granularity might affect the outcome of simulations, we designed and developed a prototype system of three infrastructure simulation networks that were then combined into one SoS simulation using High Level Architecture (HLA) implementation. We then performed a series of experiments to investigate the response of the simulation to varying time granularities. Our examination included a propagation of disruptions among constituent simulations to estimate how this was affected by the frequency of updates between those simulations, i.e. time granularity.

Our results revealed that the size of the simulated disruption decreased with in-creasing time granularity and that the simulated recovery time was also affected. In conducting this project, we also identified several ideas for future research that focus on a wider range of disruption generators and infrastructure systems in those SoS simulations.

Keywords: Time Granularity, Infrastructure System, System-of-Systems (SoS), High Level Architecture (HLA), Interdependency Study, Synchronization.

1 Introduction

As technology advances, infrastructure systems are becoming more interdependent, requiring inputs and outputs from and to other systems so that their functions are performed properly. At the same time, disruptions to these systems are increasing both in frequency and in the extent of the impact. This is especially visible within the context of urban settings, where various intertwined systems must work perfectly to ensure smooth operation of systems nearby [1]. Consequently, designing these systems to be highly resilient and ensuring that they respond adequately to any disruptions are major concerns that must be addressed. Planners and managers have to recognize how disruptions that emerge in one system can affect other systems. Currently, however, there is a lack of understanding about how time granularity in distributed modeling environments affects the outcomes of simulation experiments, including the propagation of disruptions between constituent systems.

Although models are being developed to examine interdependencies among infrastructure systems [2-5], they have not addressed the issue of time granularity. Their applications include models of traffic simulation [6] or networks of infrastructure systems [7]. However, those models are not designed to be accurate when investigating the impact of disruptions on infrastructure. There exists a model simulating interactions between infrastructure systems under disruption [8]. However, in this model the individual infrastructure systems are not run independently of each other, and hence time granularity of synchronizing the constituent infrastructure systems simulations is not analyzed.

The objective of our study was to conduct prototype experiments that assessed the impact of time granularity on the propagation of disruptions between systems in a system-of-systems (SoS) simulation. Specifically, we examined whether simulating the same event among the same infrastructure systems could produce widely differing results under various time granularities. We limited our experiments to one abstract set of infrastructure system networks and a single event rather than exploring a wider range of events. Therefore, our test results would not necessarily apply to any particular real-life system.

2 Model development

Frameworks and methodologies have been established to model individual infrastructure systems. Such infrastructure systems include power supply grids, transportation networks, water supply networks, emergency services, financial systems etc. The SoS approach can be used to illustrate interactions among interdependent systems [7-9]. In such models, the constituent systems operate on their own, being guided by their unique mechanics, but can be combined to exchange certain information (data) based on their interdependencies. In this approach, autonomous infrastructure systems interact with each other, users, operators, observers, and disruption generators. All of these components can be represented as individual systems within the overarching SoS simulation. Synchronization of these systems means that interdependencies between them are encoded in the simulation design.

One framework used for modeling SoS is High Level Architecture (HLA) [10-11], which originated in military applications. However, HLA can also be used with civil infrastructure systems [7]. This framework has three components. First, the Interface Specification determines how and where constituent systems within HLA, so-called ‘federates’, communicate with the real-time infrastructure (RTI). The second component, the Object Model Template (OMT), defines what information is exchanged between constituent simulations. As the third component, Rules specify what the federates must obey if they are to comply with the HLA overarching simulation, or ‘federation’.

Within the context of SoS simulations, federates can be infrastructure systems, operators, observers, disruption generators, or patterns of user services. These independently operating federates are connected with the RTI to exchange data and form an HLA federation. Such a system is shown in Figure 1. In general, this framework can work with any simulation methodology that includes exchange of information with each other. However, in the context of our model, the simulation programs are network simulations that correspond to infrastructure systems networks. These networks can be adaptive to the variable nature of infrastructure systems that they represent.

Although the HLA is useful for depicting the scope and method by which information is exchanged between constituent systems, it does not solve the issue of finding an adequate time granularity for the SoS. Time granularity defines when the exchange of information and, thereby, inter-system synchronization happens. Specifications for HLA include time management [12-13], which ensures that timing between federates is synchronized. However, the issue of how often to synchronize constituent federate simulations remains unanswered because it can vary between different types of simulation.

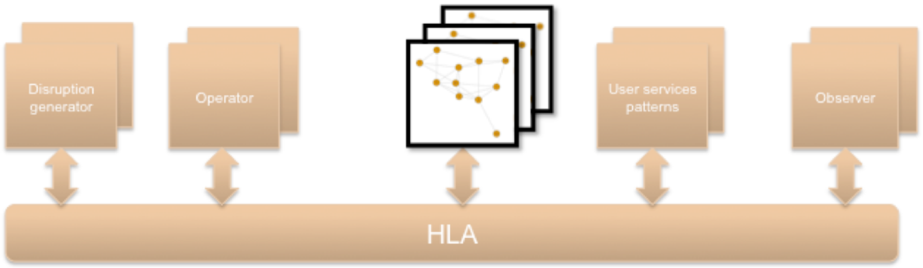

Fig. 1. HLA system-of-systems, showing components connected to real-time infrastructure to ensure exchange of information.

Time granularity is crucial when disruptions of a particular size and at a particular time are introduced by a generator into the simulation. These entry points and sizes, as well as the effects of such disruptions, can vary greatly, depending upon the time granularity of the simulation. We developed a prototype SoS of infrastructure system networks that could operate both independently and interdependently with each other. The prototype utilized an observer that could visualize the results and progress of the simulations, along with evaluating system performance. In this study, we also included a disruption generator and varied time granularity to learn how different granularities might influence the outcomes when the rest of the parameters of the simulation were held constant. These collected data included the size of the disruption and the length of recovery time.

In each network, nodes correspond to units that perform operations and interact with other nodes in the same network as well as with corresponding nodes in other federates. Edges correspond to transfer links between operational units within each network. The mechanics of each network differ and are abstract. Although each node has its own intrinsic, internal performance, it also takes inputs from the incoming edges of its network and from its corresponding nodes in the other two networks. These conditions are then combined and transformed to determine the total performance of that particular node. Performance is then propagated to the next nodes through outgoing edges. In doing so, we designed a working process for each constituent federate to simulate an individual infrastructure system network.

We also connected a disruption generator to the HLA RTI, following Poisson processes to introduce disruptions into one of the constituent networks by communicating the message that certain nodes had been affected and were to be incapacitated. This reduced to 'zero' the intrinsic performance of the affected nodes. The framework components were linked with the HLA RTI to allow for the exchange of information at synchronized points. After a certain time period, recovery occurred, the disruption was removed, and performance by those nodes was restored to its original level.

2.1 Model implementation

Our simulation was developed in C++ and Python, and the HLA modules were conceived in C++ v11 [14] through Portico 2.0.2 HLA implementation to define the interfaces between infrastructure systems, communication between them at various time granularities, and communication with the disruption generator. For graph operations, we used igraph library version 0.7.1 [15]. The disruption generator was also developed in C++ v11. Infrastructure systems were created in Python 3.5 [16], under an Anaconda 2.4.0 distribution [16], with the aid of the igraph library for graph operations and representation, and also with numpy library version 1.10.1 [18] for linear algebra and numerical operations. All results were observed with an interface web page developed in JavaScript and HTML, using the CanvasJS [19] library to visualize the performance of the simulation. This SoS simulation was evaluated and run on a Mac OS Yosemite 10.10.5 operating system.

Implementation of the system followed a natural, logical pattern. First, the infrastructure system networks and their required inputs and outputs were developed in parallel with the disruption generator and observer. Afterward, interfaces with the HLA RTI were created for each of the components.

The system was evaluated and tested with several sets of parameters and different constituent networks under varying time granularities. It performed well and was able to mimic certain real-life scenarios. We then conducted multiple tests with positive outcomes to ensure that the system propagated and communicated disruptions as expected.

3 Simulation experiments

We measured the impact of time granularity by running a small-scale simulation of three interdependent infrastructure system networks (Figure 2). Each network consisted of a certain number of nodes connected with edges. These nodes had equivalents in other networks with which they communicated by exchanging information at synchronization points. The experimental networks had the following parameters: 1) Network 1, 20 nodes connected with 75 edges; 2) Network 2, 21 nodes connected with 77 edges; and 3) Network 3, 22 nodes connected with 77 edges. They corresponded to real-life infrastructure systems such as a power grid, water supply, or transportation. The interdependencies of those networks were defined according to how each utilized the information exchanged among them.

In our simulation, a Poisson generator was used to introduce disruption into the system. The process of disruption and recovery, and the current state of the system under simulation, was examined in real-time by an observer connected to the system.

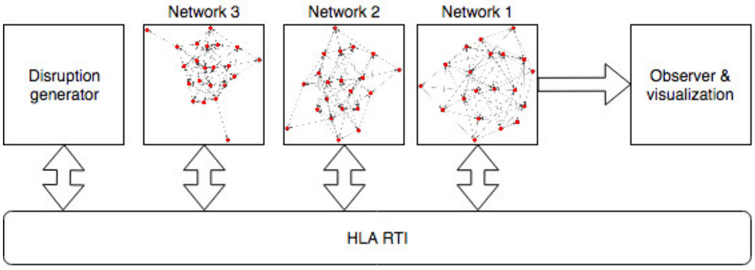

Fig. 2. Setup of experimental system involving 3 network simulations and disruption generator connected through HLA RTI. Impact of disruption on Network 3 was simulated. Propagation and related impacts were evaluated by observer and visualization module connected to Network 1.

3.1 Experimental layout

Our goal was to determine whether time granularity has an impact on simulation outcomes. To achieve this, we kept all of the simulation parameters constant (single-factor experiment) and varied only the granularity factor across 12 levels (Table 1). Recovery times were measured for the different granularities along with the maximum disruption.

Table 1. Single-factor layout.

	Timesteps											
Time granularity	1	2	3	6	9	12	15	18	21	24	27	30
Recovery time	18											

Various experimental scenarios were tested with the following parameters. Network 3 was disrupted, Network 1 was measured, and Network 2 was not affected. In Network

3, disruption was of size 16, corresponding to 16 nodes being incapacitated when the disruption was introduced. The recovery time was 18, meaning that the disruption was recovered after 18 time steps. That is, the affected nodes in Network 3 were restored to their normal performance after 18 steps. This system was swept through time granularities from 1 to 30, in increments of 3, but also included time granularities of 1 and 2. We measured the disruption size of Network 1 and measured its recovery time to within >99% of the original performance.

3.2 Simulation result

The results from this prototype simulation confirmed our expectations that changing the time granularity would have an impact on the outcome. After assessing the performance of Network 1, it was apparent that both disruption size and the recovery process were affected by granularity.

Disruption size. The main metric used to assess simulation performance was simulated disruption size, which was measured in Network 1 at each level of time granularity. Here, disruption size decreased as granularity increased (Figure 3). After initially decreasing rather slowly, when time granularity exceeds the actual recovery time, we learned that the disruption was not captured at all because its size fell to ‘0’.

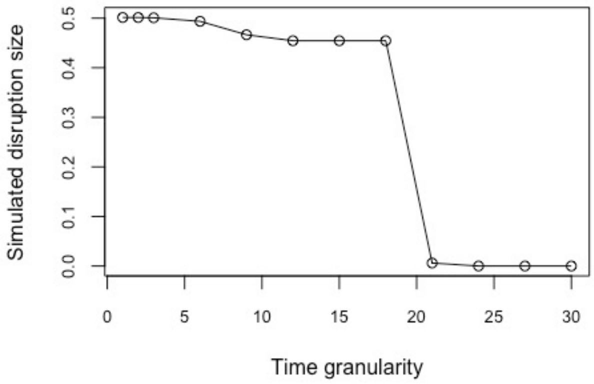

Fig. 3. Relationship between disruption size and time granularity. As granularity increased, disruption size decreased. When time granularity exceeded recovery time (18), disruption was no longer visible in simulation.

Recovery time. Simulated recovery time was expressed in time steps, beginning when the disruption was introduced in Network 1 and ending when 99% performance level was again reached. This metric allowed us to see which simulated recovery time was the closest to the actual recovery time, i.e., 18. Figure 4 illustrates how the simulated recovery time continued to increase until it arrived at half of the actual recovery time. At that point, the simulated recovery time declined before rising again, at a slower but constant rate, until the actual recovery time was reached. At a point higher than the actual recovery time, the simulated recovery time became ‘0’ because the system no longer recognized the disruption. That is, Network 3 was repaired before the disruption could propagate to other networks where it would have been registered. This scenario presented a potentially large issue when defining an adequate time granularity in a simulation.

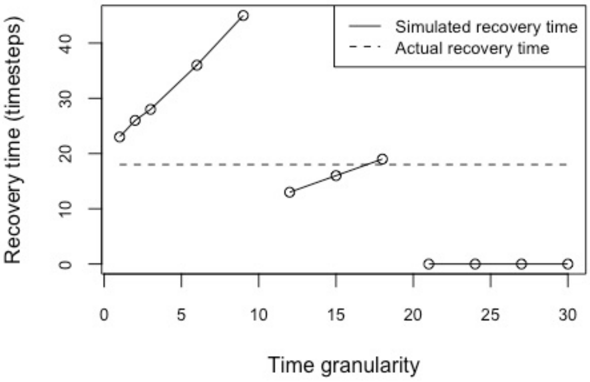

Fig. 4. Relationship between simulated recovery time and time granularity, based on time required for performance to return to 99% of original level when compared with actual recovery time. Three simulated regimes were revealed: up to half of actual recovery time (1-9), from half to actual recovery time (9-18), and above actual recovery time (18-30). The second regime proved to be most representative of actual recovery time.

4 Conclusions

Our study objective was to run simulation experiments with three federates and a disruption generator that would mimic the interdependencies among several infrastructure systems. We then investigated how time granularity might affect the outcome of those simulations.

These experiments yielded the following major results:

- As time granularity increased, the simulated disruption size was decreased.
- As time granularity increased, the simulated recovery time increased at different rates within three distinct granularity regimes: up to half of the actual recovery time, from half up to the actual recovery time, and above the actual recovery time.

- When time granularity exceeded the actual recovery time, the disruptions were no longer registered in the system.
- Time granularity has to be smaller than the actual recovery time of key events in order for the propagation of those events to be registered and visible in a SoS simulation.

Our analysis of how time granularity influences the simulation of infrastructure systems is novel. Although time management of SoS simulations has already been investigated within the context of HLA simulations [12][20], no one had previously attempted to examine time granularity within the context of modeling interdependent infrastructure systems. Although Eusgeld and Nan [3], Rinaldi [21], and Dubaniowski and Heinimann [8] had studied interdependent infrastructure systems, their research did not consider the impact of time granularity on such modeling. In the model described by Dubaniowski and Heinimann, individual infrastructure systems are combined together in one, large multi-layer network simulation, rather than multiple individual independent, distributed simulations connected together to exchange information. Thus time granularity of synchronization cannot be investigated adequately in their model.

Similarly, HLA has been utilized in modeling various SoS simulations, e.g., aircraft [22], defense [9] [23], or individual infrastructure [24]. However, none of the earlier research had introduced the concept of disruptions in those evaluations. Therefore, the results of our innovative experiments will assist scientists in developing better models of infrastructure and investigating how disruptions can affect the interdependencies of infrastructure systems.

One limitation, revealed in our tests, was the incomplete exploration of the experimental space because we varied only time granularity without considering other factors. In addition, the nature of our included networks was abstract and generally would vary with the location and type of infrastructure. To address these challenges, future studies could involve multi-factorial experiments with a greater number of variables. Moreover, real-life networks could be applied to evaluate the system within the context of a real-life SoS. Finally, one could incorporate a wider range of disruption types and generators, as well as more system networks.

Acknowledgements. This work is an outcome of the Future Resilient Systems project at the Singapore-ETH Centre (SEC), which is funded by the National Research Foundation of Singapore (NRF) under its Campus for Research Excellence and Technological Enterprise (CREATE) programme.

This is a pre-copyedited version of a contribution published in Complex Networks & Their Applications VII – Volume 1 Proceedings The 7th International Conference on Complex Networks and Their Applications COMPLEX NETWORKS 2018 published by Springer Nature Switzerland AG. The definitive authenticated version is available online via https://doi.org/ 10.1007/978-3-030-05411-3_39.

References

1. H. R. Heinimann and K. Hatfield, “Infrastructure resilience assessment, management and governance – State and perspectives,” in *Resilience and Risk*, Springer, Dordrecht, 2017, pp. 147–187.
2. C. Boer, *Distributed Simulation in Industry*. Erasmus University, Rotterdam, 2005.
3. I. Eusgeld and C. Nan, “Creating a simulation environment for critical infrastructure interdependencies study,” in *2009 IEEE International Conference on Industrial Engineering and Engineering Management*, 2009, pp. 2104–2108.
4. S. M. Rinaldi, J. P. Peerenboom, and T. K. Kelly, “Identifying, understanding, and analyzing critical infrastructure interdependencies,” *IEEE Control Systems Magazine*, vol. 21, no. 6, pp. 11–25, Dec. 2001.
5. M. Ouyang, “Review on modeling and simulation of interdependent critical infrastructure systems,” *Reliability Engineering & System Safety*, vol. 121, pp. 43–60, Jan. 2014.
6. U. Kelin, T. Schulze, and S. Straßburger, “Traffic simulation based on the High Level Architecture,” in *Proceedings of the 30th Winter Simulation Conference (WSC)*, Los Alamitos, CA, USA, 1998, pp. 1095–1104.
7. I. Eusgeld, C. Nan, and S. Dietz, “‘System-of-systems’ approach for interdependent critical infrastructures,” *Reliability Engineering & System Safety*, vol. 96, no. 6, pp. 679–686, Jun. 2011.
8. M. I. Dubaniowski and H. R. Heinimann, “A framework modeling flows of goods and services between businesses, households, and infrastructure systems,” in *Resilience The 2nd International Workshop on Modelling of Physical, Economic and Social Systems for Resilience Assessment : 14-16 December 2017, Ispra*, Ispra, Italy, 2017, vol. I, pp. 182–190.
9. J. Gao, S. V. Buldyrev, H. E. Stanley, and S. Havlin, “Networks formed from interdependent networks,” *Nature Physics*, vol. 8, no. 1, pp. 40–48, Jan. 2012.
10. J. S. Dahmann, R. M. Fujimoto, and R. M. Weatherly, “The Department of Defense High Level Architecture,” in *Proceedings of the 29th Conference on Winter Simulation*, Atlanta, GA, USA, 1997, pp. 142–149.
11. O. Topçu and H. Oğuztüzün, “High Level Architecture,” in *Guide to Distributed Simulation with HLA*, O. Topçu and H. Oğuztüzün, Eds. Cham: Springer International Publishing, 2017, pp. 29–78.
12. R. M. Fujimoto, “Time management in The High Level Architecture,” *SIMULATION*, vol. 71, no. 6, pp. 388–400, Dec. 1998.
13. R. M. Fujimoto and R. M. Weatherly, “Time management in the DoD High Level Architecture,” in *Proceedings of the Tenth Workshop on Parallel and Distributed Simulation*, Washington, DC, USA, 1996, pp. 60–67.
14. N. M. Josuttis, *The C++ Standard Library: A Tutorial and Reference*. Addison-Wesley Professional, 2012.
15. igraph.org, “python-igraph 0.7.1,” 04-Feb-2014. [Online]. Available: http://igraph.org/2014/02/04/igraph-0.7-python.html. [Accessed: 07-Aug-2018].
16. Python.org, “Python Release Python 3.5.0,” *Python.org*, 13-Sep-2015. [Online]. Available: https://www.python.org/downloads/release/python-350/. [Accessed: 07-Aug-2018].
17. Anaconda Inc., “Anaconda 2.4.0 for Python 3.5,” 02-Nov-2015. [Online]. Available: https://docs.anaconda.com/anaconda/packages/old-pkg-lists/2.4.0/py35. [Accessed: 07-Aug-2018].
18. SciPy.org, “NumPy Reference — NumPy v1.10 Manual,” 18-Oct-2015. [Online]. Available: https://docs.scipy.org/doc/numpy-1.10.1/reference/. [Accessed: 07-Aug-2018].

19. Fenopix, Inc., “CanvasJS 1.8.5 beta,” *CanvasJS*, 2013. [Online]. Available: https://canvasjs.com/. [Accessed: 07-Aug-2018].
20. R. E. J. Jansen, W. Huiskamp, J. J. Boomgaardt, and M. Brasse, “Real-time scheduling of HLA simulator components,” *TNO Fysisch en Elektronisch Laboratorium*, Jan. 2004.
21. S. M. Rinaldi, “Modeling and simulating critical infrastructures and their interdependencies,” in *Proceedings of the 37th Annual Hawaii International Conference on System Sciences, 2004*, Big Island, HI, USA, 2004.
22. C. Gervais, J.-B. Chaudron, P. Siron, R. Leconte, and D. Saussie, “Real-time distributed aircraft simulation Through HLA,” in *Proceedings of the 2012 IEEE/ACM 16th International Symposium on Distributed Simulation and Real Time Applications*, Washington, DC, USA, 2012, pp. 251–254.
23. S. L. Ferenci, M. Choi, J. Evans, R. M. Fujimoto, C. Alspaugh, and A. K. Legaspi, “Experiences integrating NETWARS with the naval simulation system using the high level architecture,” in *IEEE MILCOM 2004. Military Communications Conference,* vol. 3, 2004, pp. 1395–1401.
24. A. Jain, D. Robinson, B. Dilkina, and R. Fujimoto, “An approach to integrate inter-dependent simulations using HLA with applications to sustainable urban development,” in *2016 Winter Simulation Conference (WSC)*, Washington, DC, USA, 2016, pp. 1218–1229.