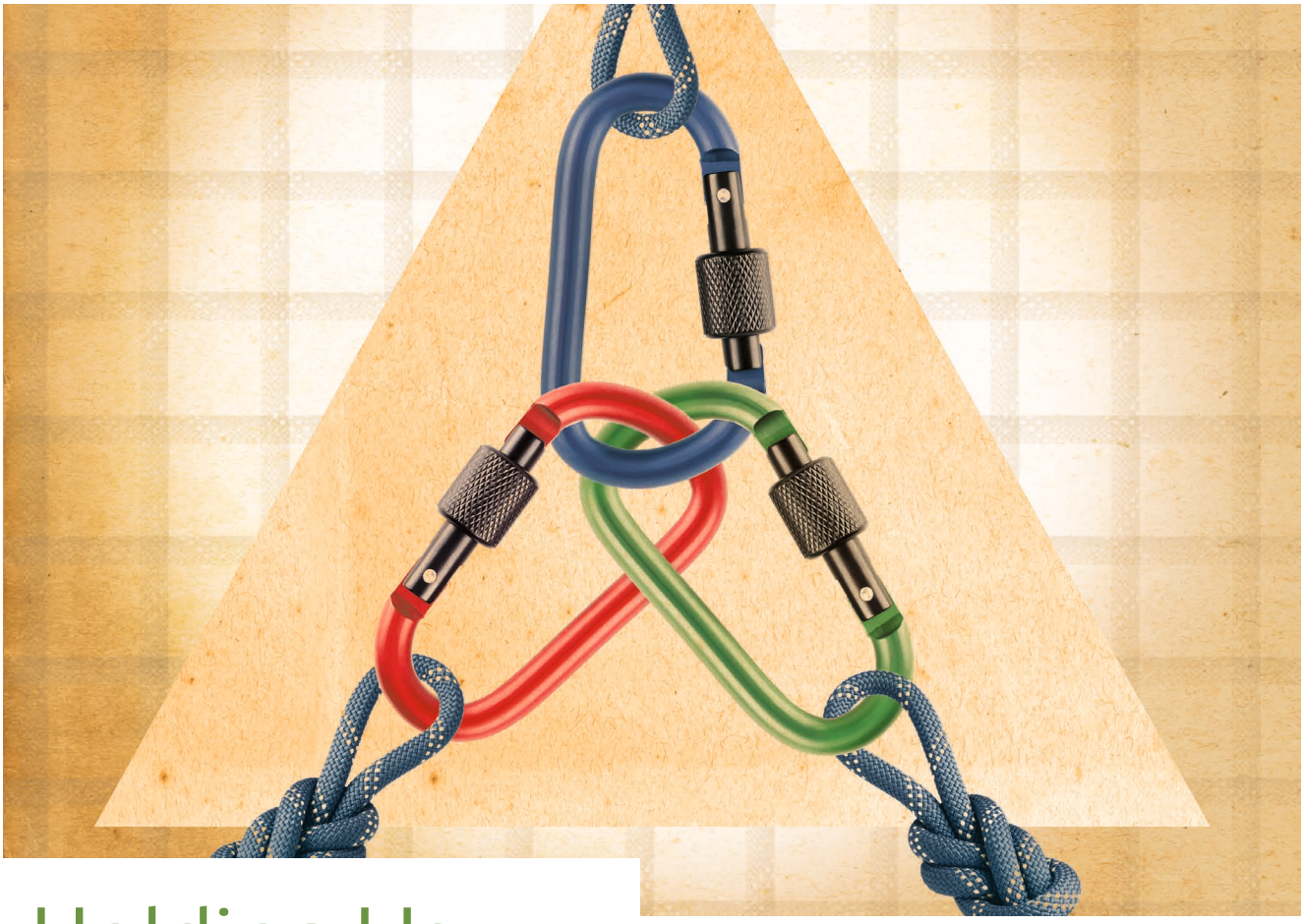
EDITOR IN CHIEF **DAVID ALAN GRIER**
Djaghe, LLC; dagrier.dc@gmail.com



Holding Us Together

David Alan Grier, Djaghe, LLC

The articles in this issue draw from the past, offer a bit of present insight, and look to the future. Taken together, you get a sense of what the entire field of computing is doing today and how it all holds together.

The fields of computer science, computer engineering, and software engineering are remarkably broad, and few people can fully encompass all three. Fewer still can clearly see how these fields interact with each other, how each borrows from the other and comments on a different approach to the problems of computation, and how they work together as a complex whole. In this issue, we present five articles that demonstrate the diversity of computation and point to the common goal of making computation useful.

If I were to identify a good entry point for this issue, I might suggest Antinyan, Sandberg, and Staron's article on code complexity, "A Pragmatic View on Code Complexity Management." The authors identify an important point in this article: There is a big difference between complexity that is essential and complexity that is a byproduct

Digital Object Identifier 10.1109/MC.2019.2897033
Date of publication: 22 March 2019

 0018-9162/19©2019IEEE

of the engineering process. All of us are engaged in the work of managing complexity and ensuring that our systems have enough elements to work properly but not so many that they become obscure or brittle.

Since complexity is often the enemy of agility and robustness, you may want to jump next to "Software Systems With Antifragility to Downtime" by Hole and Otterstad, which discusses how to think about systems that are robust and should operate continually. These authors approach the subject from a negative point of view when they write about "software systems that are antifragile against downtime." As a good editor, I normally would highlight the text and tell them to phrase their results in a positive way. However, they have good reason to approach the subject from a negative viewpoint, since antifragility really is more of a strategy for building good systems and avoiding the problems that bring software to a stop.

Failed systems, through either complexity or fragility, can indicate a failed engineering process and invite a legal investigation. However, there often are legal implications for software that is functioning properly. In "Algorithms: Law and Regulations," Treleaven, Barnett, and Koshiyama ask if there is a new legal status for algorithms and if we need to start thinking about how algorithms interact with legal institutions.

Legal structures are important because we don't always see the implications of our work. Siddique, Akhtar, Khan, and Kim, the authors of the next article, "KDD Cup 99 Data Sets: A Perspective on the Role of Data Sets in Network Intrusion Detection Research," look at a classic problem of computer science that was proposed at the 1999 Knowledge Discovery and Data Mining Conference (KDD). At that conference, DARPA proposed a contest to build an algorithm that could detect potential intruders from the patterns in network traffic. (The contest and its result are in IEEE *Xplore* at https://ieeexplore.ieee.org/document/821515.) This current article suggests that, while the contest may have been important to the KDD field, its data may not be the best example of current intruder traffic.

The last article, by Bresniker et al., "Rack-Scale Capabilities: Fine-Grained Protection for Large-Scale Memories" returns to the hardware roots of computing but does so with a modern twist. It looks at the problem of properly securing networked memory units, particularly the fabric-attached memory that has been part of Hewlett-Packard's research on new network articles. Like all of the articles here, this one presents an interesting solution, identifies the strengths and weaknesses of that solution, and suggests the next steps for research.

Of course, that is the common feature of the articles in this issue. They draw from the past, give some insight into the present, and look to the future. That is what the entire field of computing is doing today.

DAVID ALAN GRIER is a principal with Djaghe, LLC. He is a Fellow of the IEEE. Contact him at dagrier.dc@gmail.com.

Digital Object Identifier 10.1109/MC.2019.2901912