Introduction: Software Dependability

Long Wang

Abstract This is the introduction of the 6 chapters in this "software dependability" section. Threats to software dependability are getting aggravated as more complex software and systems are being used and hardware devices with thinner MOSFET channel lengths are being used. This section presents 6 state-of-the-art work that demonstrate a few trends in software dependability research: popular use of data-driven AI, blurring limits between software dependability and security, and software dependability and security in emerging computing environments. The audience will get an up-to-date view of the software dependability research, especially its ongoing trends, after reading this section.

Keywords Dependability · Security · Blurring limit

Information technology (IT) is rapidly expanding its application scope and spreading into more critical domains such as electric power management, transportation traffic regulation and public health, in addition to the traditional domains of scientific computing, office business, finance and telecommunication, etc. Large computing platforms such as cloud systems and artificial intelligence (AI) platforms, and large networks such as internet-of-things (IoT) network are emerging as key computing infrastructures that host IT services. As a result, the complexities of software running on these modern computing systems have been increasing by a lot.

The rapid spread of software into broader critical domains and the increasing complexities of software demand high dependability of software. Moreover, hardware devices underlying computing systems are using MOSFET (or similar technologies) devices with very thin channel length (5 nm, or thinner expected in near future), which give rise to a much larger amount of soft errors in computing systems. This issue further aggravates the software dependability problem, and demands more focus be placed on software dependability in modern computing systems. However, the rapid progress of IT technologies also brings new capabilities of improving software dependability.

L. Wang (✉)
Tsinghua University, Beijing, China
e-mail: longwang@tsinghua.edu.cn

L. Wang et al. (eds.), *System Dependability and Analytics*, Springer Series in Reliability Engineering, https://doi.org/10.1007/978-3-031-02063-6_1

This section presents a select set of state-of-the-art work that demonstrate a few trends in software dependability research now. (i) One recent principal thrust addressing software dependability is through data-driven AI, including machine learning based on deep neural network, data analytics, and various classification techniques. (ii) Another trend is the blurring limits between software dependability and software security. Specifically, a number of technologies originally proposed and traditionally applied for software dependability are recently applied for software security and have demonstrated their significance in addressing security issues. Examples include bit flip injection, fuzzing (exploration of various inputs for tests), formal method, distributed consensus and monitoring technologies. As the limits between software dependability and security get blurring a new gate is open, and a number of technology advancements are being proposed and then employed in practice. (iii) Software dependability and security in emerging computing environments such as cloud systems and IoT environments are also hot topics recently.

The first two articles of this section demonstrate two good examples on how data-driven AI is adopted for addressing software dependability issues. *Intelligent Software Engineering for Reliable Cloud Operations*, authored by Prof. Lyu and Prof. Su, describes an AIOps (Artificial Intelligence for IT Operations) framework that employs AI technologies for anomaly detection in cloud systems. The framework leverages existing monitoring data of a cloud, particularly Key Performance Indicators (KPIs) data such as CPU usages of VMs, packet loss rates, packet error rates, etc., and applies neural network models to do anomaly detection and generate system incidents. Then the framework applies Graph Representative Learning algorithms to cluster and aggregate the incidents for failure diagnosis and root cause analysis. Hanmer and Prof. Mendiratta's *Data Analytics: Predicting Software Bugs in Industrial Products* presents a survey of software bug prediction techniques and a case study that employs source code complexity metrics, such as percent branch statements, block depth, line number of deepest block, statements at block level 0, to do bug prediction. The proposed technique in the case study uses Random Forest for the prediction. The two articles show that AI has demonstrated its super powerful capabilities in identifying patterns in complicated data, and such capabilities greatly help with anomaly detection, failure diagnosis, and error prediction.

The following three articles are examples that show blurring limits between software dependability and software security. Dr. Chen's *From dependability to security—a path in the trustworthy computing research* provides enlightenments on the relationships between dependability and security, between faults and attacks, by virtue of the author's own experience. Dependability and security are discussed in context of a common adversary model. Particularly, "bit flips", "formal methods" and "distributed consensus" are discussed as the main instruments used for both dependability and security (actually most of them, if not all, were proposed and applied first for dependability, and then repurposed for security). *Assessment of Security Defense of Native Programs Against Software Faults* by Dr. Yim studies security defense of C/C++ programs against faults. Faults and attacks, though they are two distinct adversaries of programs, are related in that faults, e.g. bit flips, may cause consequences of security breaches. This article conducts experimental studies of

"exploitable software faults", the software faults that can be exploited to result in security breaches, and shows both the capability of the fuzzing technology in finding exploitable software faults and the built-in security defense capability of programs against exploitable software faults. The article exposes interesting insights on how security-oriented exploitation and reliability faults are related. *Multi-layered Monitoring for Virtual Machines* by Dr. Pham describes a solution of VM monitoring for both reliability and security purposes. The solution covers all layers from hardware and hypervisor up to applications. It provides a quite comprehensive description of VM monitoring technologies. The audience will understand the challenges, pros and cons of VM monitoring technologies after reading this article. The three articles are part of the ongoing efforts that combine dependability research and security research.

The last article in this section, Prof. Bagchi's *Security for Software on Tiny Devices*, presents research challenges and potential approaches for providing security to software running on IoT devices. This is a very good introduction on software security on IoT devices. The unique challenges are clearly stated, and the discussions in the article span analysis techniques and algorithms, the enforcement of IoT software security that implements the analysis techniques and algorithms, and measurements, metrics and evaluations of IoT software security. The audience will obtain a clear view of state-of-the-art of the IoT software security from the article.

In summary, this section focuses on software dependability and presents a select set of state-of-the-art work on it. The audience of the section will get an up-to-date view of the software dependability research, especially its ongoing trends. This view is very important today as software dependability is gaining an unprecedented demand while undergoing a drastic change. Both are brought about by the wide and rapid adoption of technology advancements in cloud computing, AI, and other areas: IT services (and software) are growingly supporting more applications and scenarios including many in the critical domains such as public health, transportation traffic regulation and driving of vehicles, where traditionally IT technologies were not largely involved; at the same time, the technology advancements give rise to new approaches, many drastically different from traditional ones, to addressing software dependability issues.