

Chapter 1

Introduction

Modern societies cannot function without information and communications technology (ICT) systems. When ICT systems such as electronic government (e-government) systems, e-payment infrastructures, and mobile phone networks fail, users can still access alternative systems based on older technologies, but these alternatives are rapidly disappearing. E-government services are introduced on the Web to improve services to citizens and to free up the human resources needed to tackle the increasing health care requirements of aging populations. Since there will not be enough government employees to handle a large number of requests over the phone, on paper, or by personal appointment in the future, it is necessary to deploy e-government services that remain robust to undesirable incidents over time and that are available to citizens around the clock.

The robustness and availability of national e-payment infrastructures are also becoming increasingly important as countries are becoming cashless societies. If a nationwide e-payment infrastructure goes down in a cash-free society, people will not be able to pay for necessities. In Scandinavia, this is already more or less the case, since many people, especially the young, no longer carry cash. While Norwegian banks want to abolish cash altogether to reduce costs, there have been enough incidents over the last ten years causing unplanned downtime and erroneous account withdrawals to question whether the current e-payment infrastructure can provide the very high availability and long-term robustness required by a completely cash-free society.

Mobile phone networks have nearly replaced fixed-line phone systems in many countries. It is difficult to find spare parts for the old landline systems and they are expensive to maintain. Norway's largest telecom company wants all remaining fixed-phone subscribers to move to mobile subscription plans so it can dismantle the landline system altogether. Several large incidents have demonstrated how dependent the Norwegian population has become on mobile phone networks. When areas on the west coast of Norway lost power for several days because of a severe storm, the local inhabitants mainly complained about the mobile phone networks being down, illustrating that people now expect their mobile phones to work anytime and anywhere.

As traditional governmental services, cash-based payment systems, and landline phone networks are disappearing, there is a growing need for very large ICT systems with very high availability and sustained robustness to unwanted incidents. How should such systems be designed and operated to meet the increasing expectations of users in a rapidly changing world? Is a particular system design fragilizing a service of importance to millions of users? Will users be exposed to incidents with intolerable impact? *Common mode failure* is a particularly important challenge, defined as a failure in multiple parts of a system due to a single event. How do we prevent single events from propagating and taking down many parts in the same manner? This book tries to answer these questions by modeling large ICT systems as complex adaptive systems.

1.1 Complex Adaptive Systems

The term *complex adaptive system* denotes a man-made or natural system consisting of many entities that interact in involved ways. The entities adapt to each other and the environment to enable the system as a whole to survive events with potentially large negative impact [1–7]. ICT systems consisting of large networked computer systems and many stakeholders, including users, operators, and owners, are complex adaptive systems, as illustrated in Fig. 1.1. The complexity is due primarily to the numerous interactions between the stakeholders and the computer systems, the large amounts of communications between the networked subsystems, and the influence of changing security and privacy policies, as well as threats such as equipment failure, extreme weather, and sabotage. Collections of software services running on cloud computing platforms and nationwide infrastructures for mobile telecom constitute two particularly interesting classes of complex adaptive ICT systems with many users, mutually dependent entities, and self-regulating behaviors.

To gain an understanding of why governments and companies build complex adaptive ICT systems, we consider how valuable distributed ICT systems are to their

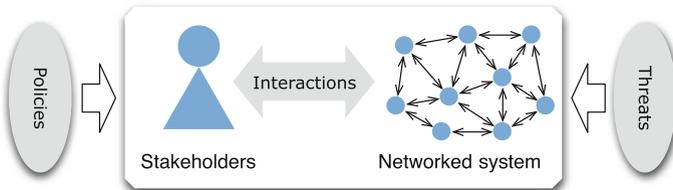


Fig. 1.1 A complex ICT system’s global behavior is caused by dynamic interactions between the stakeholders and the networked computer system and by interactions between the network’s many subsystems. Changes to policies, threats, or subsystems can cause sudden and large changes in global behavior

owners. Here, value can be the ability to provide a population with transactional services on the Web or the revenue from an online social network. Consider a system with N users. The number of possible pairs of connections between users is $N(N - 1)/2$, giving rise to Metcalfe's law, stating that the value of a system is proportional to the square of the number of connected users, N^2 . Alternatively, there are $2^N - N - 1$ possible sub-groups of users, resulting in Reed's law, stating that the value of a system scales exponentially with the number of users, 2^N . Both laws indicate that the value of distributed ICT systems grows very rapidly with the number of users, making it desirable for governments and companies to build huge ICT systems of high complexity. In addition, for many networked systems, every new user makes a system's services more valuable to the other users.

Complex adaptive systems contain *feedback loops*, as illustrated in Fig. 1.2. A feedback loop is a series of interacting processes that together result in a system adapting to the effect of its previous behavior. Feedback loops are what make complex systems adaptive. The loops create emergent global patterns or behaviors. *Positive* (escalating or compounding) feedback loops propagate and turn local events into global events, affecting whole systems, while *negative* (dampening or stabilizing) feedback loops limit the impact of local events affecting parts of systems. Negative feedback typically stabilizes a system's global behavior over a certain operating range, while positive feedback creates extreme global behavior outside the normal operating range [1, 3, 4]. Ideally, complex adaptive ICT systems should prevent positive feedback loops from ever propagating local failures into extreme global behaviors and causing systemic failures.

The emergent global behaviors of complex adaptive ICT systems are often modeled as stochastic events with given probability distributions. We distinguish between thin-tailed and thick-tailed distributions (see Chap. 2). If the tails are thin, then outliers in the form of extreme global behaviors can be ignored because the thin tails make the outliers very unlikely. When the distributions have thick tails, the outliers cannot be ignored because the probability that at least one outlier will occur is significant. Many man-made systems, including ICT systems, have positive feedback loops that cause certain local events to propagate and create extreme global behaviors. The extreme behaviors, especially unplanned downtime, become more common than stakeholders can accept. These outliers are modeled by probability distributions with thick tails. Unfortunately, classical methods for risk analysis based on predictions of

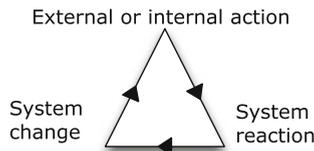


Fig. 1.2 Generic feedback loop: an external or internal action leads to a system reaction. The reaction then causes the system to change, which initiates another action and the process repeats

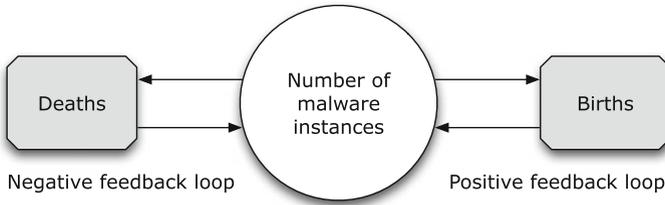


Fig. 1.3 Birth–death process illustrating malware spreading

future events tend to underestimate or ignore extreme global behaviors in complex adaptive ICT systems, even though these events may very well dominate the overall risk to stakeholders.

A vulnerability in a man-made system can be a flaw in the design, a bug in the implementation, or a mistake in the system’s operation or management. Any complex man-made system has vulnerabilities. Coincidental errors and malfunctions, as well as hostile and targeted attacks, exploit vulnerabilities to cause failures leading to extreme global behavior such as unplanned system downtime. In particular, malicious software, or *malware*, can exploit vulnerabilities and cause information leakage. Figure 1.3 depicts a simple model of an infectious malware epidemic that involves a positive feedback loop of increased births and a negative loop of increased deaths. Without deaths, the population size will increase exponentially, that is, negative feedback is needed to keep the positive feedback under control [1].

The observed fragility of complex ICT systems to prolonged downtime and malware infections demonstrates the need for better system design, implementation, operation, and management. The many interactions between the adaptive entities in the systems create a highly non-linear and time-varying relation between the input and output that makes it nearly impossible to predict extreme global behavior. Hence, we need non-predictable techniques to create complex adaptive ICT systems. Taleb’s work [8–12] suggests that we should develop and operate so-called *anti-fragile* systems characterized by two important properties: First, an anti-fragile ICT system fails early with a small, local impact to break positive feedback loops before they can create extreme global behaviors. Second, the prevention of extreme global behaviors allows stakeholders to learn from small-impact incidents about new vulnerabilities caused by changes in the system and its environment. The vulnerabilities can then be mitigated to avoid future extreme behaviors.

This book investigates how to develop and operate anti-fragile ICT systems. Cloud-based systems are emphasized because cloud computing platforms utilizing virtualization technologies greatly facilitate the creation and maintenance of anti-fragility compared to traditional datacenters without virtualization technologies (see Chap. 5).



Fig. 1.4 Toward anti-fragility

1.2 Fragile, Robust, and Anti-fragile Systems

The research literature has long categorized complex adaptive systems as fragile or robust to incidents with a particular type of impact. Fragile systems are vulnerable to the impacts of these incidents, while robust systems withstand or absorb them. Unlike robust systems, anti-fragile systems learn from such incidents how to function increasingly well in a changing environment [9, 10]. In fact, anti-fragile systems need incidents to remain well adapted to their environments. Without the ability to learn from incidents, anti-fragile systems become fragile over time as the systems themselves and their environments change. The human immune system, with its ability to adapt and self-repair, is a prime example of a system that is anti-fragile to many types of impact.

As depicted in Fig. 1.4, the fragility, robustness, and anti-fragility to a particular type of impact are best viewed as degrees on a spectrum with fragile systems to the left, robust systems in the middle, and anti-fragile systems to the right. Systems have to become robust before they can become anti-fragile and no system can be anti-fragile to all possible types of impact [10]. At present, there is no general technique to measure fragility, robustness, and anti-fragility. However, this book will demonstrate that it is not difficult to recognize when a system is fragile, for example, to downtime or malware spreading. Furthermore, it will introduce design and operational principles that move toward anti-fragility in Fig. 1.4.

1.3 Overview of Book

Taleb [10] introduced the concept of anti-fragility to analyze and explain why it is not enough for large natural or man-made systems to be robust to predictable events with large impact. In an unpredictable world, systems must be able to handle randomness, volatility, and unforeseen large-impact events. Learning from incidents is needed to prevent systems from developing fragilities over time.

So far there are no general methods or theories on how to develop or operate anti-fragile ICT systems. This book studies select philosophical and practical aspects of anti-fragile ICT systems to gain an initial understanding of them. The main message is that we should stop building fragile ICT systems of national or international importance and start building anti-fragile ICT systems.

The book is divided into five parts. Part I discusses the concept of anti-fragility, why the concept is important, and how to achieve anti-fragility in general. Part II

outlines in some detail how different ICT systems can achieve anti-fragility to downtime and Part III develops a technique to achieve anti-fragility to malware spreading. Since we need to detect failures to achieve anti-fragility, Part IV discusses how to detect anomalies in system behavior. Finally, Part V summarizes the book's main insights and suggests potential venues for further work.

The contents of Parts I and II should be easy to understand for most readers, while an additional effort may be needed to understand the more complicated content of Parts III and IV. To facilitate understanding, certain chapters repeat central information introduced earlier in the book. The following sections provide more detailed summaries of the five parts.

1.4 Creating and Maintaining Anti-fragility

Part I, including the current chapter, outlines how to create and operate complex adaptive ICT systems with anti-fragility to different types of impact, such as unplanned downtime and malware spreading. Chapter 2 first discusses rare events with a large negative impact and argues that it is, at best, very hard to predict all such events in complex systems. Next, it explains why a system must limit the impact of these events to gain robustness and why learning from the remaining events with a small impact is necessary to achieve anti-fragility.

While organizations with anti-fragile systems must accept and learn from failures, they also need to focus on building trust with users to maintain and increase their user base. Chapter 3 defines a simple agent-based model of how trust changes in a user population. The model illustrates that trust is fragile to incidents directly affecting few users and that massive distrust is robust to large efforts to regain trust. Since it is very hard to predict which events have the potential to create massive distrust, organizations must have procedures in place to handle the impact of incidents before distrust starts to spread.

The design of an ICT system is the process of defining its components, interfaces, data formats, data flow, and data storage that together satisfy specified availability, performance, and scalability requirements. Chapter 4 first provides four design principles that isolate local failures, keeping their impacts small, while supporting stringent requirements. Second, it introduces one operational principle that enables stakeholders to quickly learn from natural and induced failures to maintain a level of anti-fragility as a system and its environment change. While each principle alone does not provide any new fundamental insight, collectively the five principles outline a novel way to design and operate anti-fragile ICT systems. In particular, it is possible to create ICT systems with higher availability than today's tightly connected and highly optimized systems with limited redundancy and diversity.

1.5 Anti-fragility to Downtime

A cloud computing platform enables ubiquitous, convenient, on-demand access to a shared pool of configurable computing resources such as CPUs, networks, memory, and databases that can be rapidly provisioned and released with minimal management effort or service provider interaction [13, 14]. Virtualization technologies are used to abstract the resources for applications and end users. The availability of a cloud-based ICT system is measured by the percentage of time it is accessible to users. A high availability of 99.99%, referred to as four nines of availability, corresponds to about 53 min of downtime each year. A complex ICT system is fragile to downtime if its availability is unacceptably low to some stakeholders, robust if the availability is acceptable to all stakeholders, and anti-fragile if stakeholders or the technical system itself learn to maintain an acceptable availability as the system and its environment change.

Part II discusses how the five design and operational principles from Chap. 4 can be implemented on cloud computing platforms to create highly available software solutions and mobile telecom infrastructures. To gain insight, we reason about real systems using philosophical concepts, objects, figures, and impressions of past incidents. The thinking is based solely on publicly available information and is rooted in complexity science [15–20], where whole systems cannot be understood by exclusively studying their parts; instead, it is necessary to emphasize interrelationships and changes to understand the systems' dynamic global behaviors.

Chapter 5 outlines how Netflix implemented the five principles in their cloud-based web-scale solution for media streaming. Chapter 6 explains why Norway's e-government system has experienced too much downtime and describes how a new cloud-based system founded on the five principles can achieve anti-fragility to downtime. The chapter also references the UK e-government system to argue the need for user-focused and iterative software development to achieve anti-fragility. Finally, Chap. 7 discusses fragility to downtime in Norwegian telecom systems and outlines how the five principles applied on cloud computing platforms can make telecom systems anti-fragile to downtime.

1.6 Anti-fragility to Malware Spreading

Malware is any form of malicious software used to disrupt computer operations, gather sensitive information, or gain unauthorized access to private computer systems. Malware appears as executable code, scripts, active content, and other software. Malware includes computer viruses, worms, trojans, ransomware, spyware, scareware, and other types of malicious programs. Worldwide, trojans, worms, and viruses continue to dominate among the many malware types.

Malware is a serious threat to anybody using a computer system connected to the Internet [21, 22]. A networked system is fragile to malware spreading when local

outbreaks spread far and robust when new malware outbreaks have very limited spreading. The system is anti-fragile to malware spreading if it first learns to reduce the fraction of infected devices, for example, to less than 1% and then manages to keep the fraction of infected devices low even as the spreading mechanism of the malware changes.

Part III develops a novel malware-halting technique that prevents frequent malware outbreaks from propagating over huge networks of computing devices. Calculations and simulations using slightly modified epidemiological models from network science [23] determine the time-averaged fraction of infected devices. Chapter 8 outlines how application stores utilizing compilers with so-called diversity engines [24] can generate enough software diversity to gain robustness to malware spreading by halting frequent malware outbreaks with a fixed spreading mechanism. It also argues that diversity slows down persistent targeted attacks.

Chapter 9 studies malware types that spread over networks with an unknown topology. The malware studied have the ability to reinfect nodes multiple times. Acquaintance immunization [25] and software diversity are combined to gain robustness to malware reinfections. While reinfections generally help malware stay alive for a long time, the described halting technique prevents malware outbreaks from spreading very far before they die out.

Chapter 10 combines cloud computing, time-varying software diversity, immunization, and imperfect malware detection/removal to model and analyze networks that gain anti-fragility to malware spreading by learning to halt and remove malware with unknown and time-varying spreading mechanisms. Non-infectious malware mistakenly downloaded by computer users are viewed as infectious malware with limited spreading ability.

1.7 Anomaly Detection

To achieve anti-fragility to a particular type of intolerable impact, local failures must be detected before they can propagate into systemic failures. Humans are often needed to determine whether a local anomaly is just a benign change or a local failure with the potential to create a systemic failure. Current ICT systems deploy various techniques and heuristics to detect anomalies. For example, banks and credit card companies have a rich set of heuristics to detect fraud [26].

In Part IV, we study a general learning algorithm based on the biology of the brain's neocortex. The learning algorithm was developed by Hawkins [27] and implemented in software by the company Numenta (<http://numenta.com>). The algorithm is able to predict the behavior of a wide variety of systems. If a prediction and the actual behavior differ, then an anomaly is detected. Chapter 11 discusses the biological basis for the learning algorithm and provides an overview of the algorithm itself.

Numenta's experiments with different types of streaming data, including metric data from cloud applications, show that the algorithm detects anomalies that are

hard for humans to discover. Chapter 12 illustrates how the early detection of subtle anomalies allows systems or their stakeholders to take early action to prevent local failures from creating intolerable impact.

1.8 Ongoing Explanatory Work

Part V consists of Chap. 13. It summarizes the book's main insights and discusses possible research directions to further increase the understanding of anti-fragile ICT systems.

Overall, this book is a result of the author's ongoing long-term effort to understand what Taleb's [8–12] philosophical investigations and Geer's [28–33] systems thinking tell us about the design, implementation, operation, and management of complex adaptive ICT systems. Both Geer and Taleb look to nature to understand anti-fragile systems. In nature, sexual reproduction creates many species consisting of individuals who genetically differ from each other. An infectious disease is very unlikely to wipe out an entire population, since some individuals are almost certainly genetically immune. In other words, while each individual is vulnerable to diseases, the population survives due to a diverse gene pool.

While the author avoids superficial references to biology and Darwin's theory of evolution in the book, he agrees with Geer and Taleb that there is much to learn from nature on how to build complex ICT systems. In particular, Geer has stated several times that computing devices should have a relatively short life unless they are easy to upgrade. This observation has strongly influenced the work in Part III, leading to a novel approach to malware halting.

It is hard to precisely model the global behaviors of complex ICT systems. Rather than trying to develop sophisticated models to accurately simulate the behaviors of real systems, this book develops toy models to gain an understanding of them. While these models cannot predict the global behavior of real systems, they provide explanations of important system properties. The suggested malware-halting technique in Part III demonstrates that it is possible to create novel approaches and solutions to difficult problems by developing simple agent-based models of networked ICT systems and then employing techniques from network science to analyze the models' properties.

Since many of the ideas presented in this book have yet to be tested in real systems and since the book by no means covers all aspects of anti-fragile ICT systems, the author welcomes criticism and debate to shed further light on how to develop and operate anti-fragile ICT systems. Understanding all aspects of these systems is an important task for both the research community and the industry—not only for the author.

Open Access This chapter is distributed under the terms of the Creative Commons Attribution-Noncommercial 2.5 License (<http://creativecommons.org/licenses/by-nc/2.5/>) which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

The images or other third party material in this chapter are included in the work's Creative Commons license, unless indicated otherwise in the credit line; if such material is not included in the work's Creative Commons license and the respective action is not permitted by statutory regulation, users will need to obtain permission from the license holder to duplicate, adapt or reproduce the material.