# Open Systems Dependability and DEOS:

## Concept, Retrospect and Prospects

Mario TOKORO[1]
The Association of Dependability Engineering for Open Systems (DEOS Association)
Japan

***Abstract*—Open Systems Dependability is the concept of dependability for "open systems" so as to be applicable to those systems that operate for extended periods of time while modifying themselves to accommodate change in their environments and objectives. Typical examples would be social infrastructures and their software, as well as the world of IoT that continues to develop today. The DEOS lifecycle model, also called the DEOS process, realizes the concept in a dual-cycle iterative process with concurrent component processes for achieving accountability. The concept and the process are under discussion as the basis of an international standard (IEC 62853 Open Systems Dependability). In parallel, the DEOS process is being applied to various practical applications. This paper briefly discusses the concept, retrospective, and prospects for Open Systems Dependability and DEOS.**

***Keywords—Open systems, dependability, change accommodation, failure response, consensus building, accountability achievement, IEC 62853component***

## I. INTRODUCTION

The types of system for which we must now give serious consideration to dependability include those operating for extended periods of time while modifying themselves to accommodate change in their environments and objectives. For example, many embedded systems are connected to their central servers and those servers are connected to others to form the Internet of things (IoT). Systems developed to meet certain objectives often need to accommodate new ones, and therefore, must be modified. Component systems and their connections may change for new services, better performance, or cost savings in a system of systems (SoS). In reality, our social infrastructures undergo constant change despite being expected to continue services without any critical impact on daily lives.

I postulate that those systems should be considered and treated as open systems rather than closed ones, and therefore, that we cannot choose a constructive approach, but an iterative approach to achieve dependability. Open Systems Dependability is the concept of dependability for open systems. The DEOS process realized this concept in the form of a dual cycle, comprising the Change Accommodation Cycle and the Failure Response Cycle, and two functions—namely, Consensus Building for the system among stakeholders and Accountability Achievement.

## II. OPEN SYSTEMS AND OPEN SYSTEMS DEPENDABILITY

As the name suggests, open systems are the opposite of closed systems (see Figure 1). A closed system is a system that has no interaction at all, or a fixed interaction, with the outside world. A closed system can be characterized as having a clear boundary, structure, and functions, all of which are fixed, and therefore, we can apply a reductionistic and constructive approach with it.

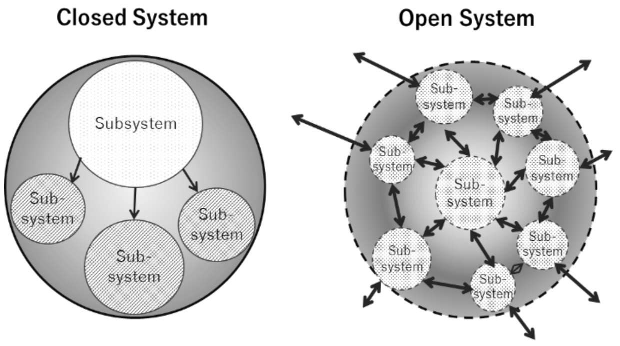

**Figure 1 Closed Systems vs. Open Systems**

An open system, on the other hand, is a system that interacts with the outside world, as defined by biologist Bertalanffy[1] in his General System Theory. Therefore, its boundary, internal structure, and functions may not be clear and may change over time. Eventually, reductionistic and constructive approaches become inapplicable and only an iterative approach is practical. A philosophical base for the method of science for open systems is found in "falsifiability" by Karl Popper[2], and a method called "Open Systems Science" has been proposed by Mario Tokoro[3][4].

The concept of Open Systems Dependability was developed in parallel with the above observations and understanding. Open Systems Dependability was defined as the ability of a system operating for an extended period of time in a real-world environment to accommodate change in its environments or objectives, to ensure that accountability in regard to the system is continually achieved, and to provide the expected services to users without interruption[5].

In order for a system to realize the definition of Open Systems Dependability, the system may itself need to provide

[1] The Institute for Open Systems Science, Ltd., `mario.tokoro@OpenSystemsScience.co.jp`

the following functions: (1) a function for modifying itself in response to the corresponding change requirements (change accommodation), (2) a function for responding to failure so as to minimize damage and continue its operation to the greatest extent possible, and also for preventing recurrence of an identical or similar failure (failure response), (3) a function for reaching agreement among stakeholders regarding the design, implementation, and operation of the system (consensus building), and (4) a function for supporting accountability of the system (accountability achievement). In order to support functions (3) and (4), the system needs to provide a facility for recording all of the agreements made and events having occurred in the system. All of these functions must be realized within the system to form an iterative process so as to be applicable to a system operating for an extended period of time.

## III. THE DEOS PROCESS

The DEOS process is configured so as to implement the functions of Open Systems Dependability described above as shown in Figure 2. The process consists of two cycles — the *Change Accommodation Cycle* (outer loop) and the *Failure Response Cycle* (inner loop) — both of which are initiated from the *Ordinary Operation state*. The *Ordinary Operation state* is the state of the system when providing normal services to users without deviating from the allowable range of service levels agreed upon by the stakeholders.

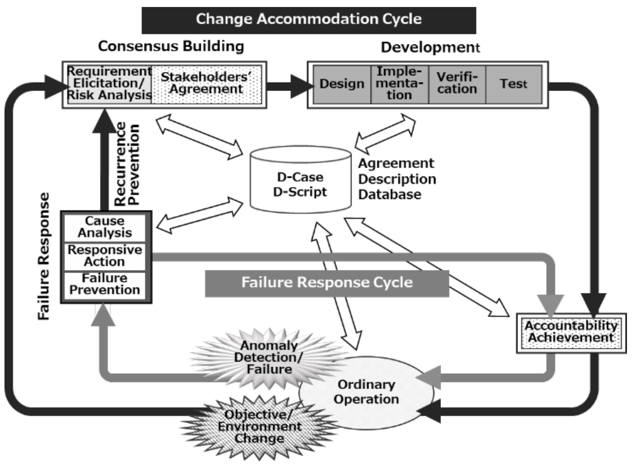

**Figure 2 The DEOS Process**

The *Change Accommodation Cycle* comprises the Consensus Building process, the Development process, and the Accountability Achievement process. It is initiated by a person, team, or committee upon recognition of a change in environments or objectives requiring the system to be modified. It can also be initiated when the *Failure Response Cycle* reports the need for system modification in order to prevent recurrence of an identical or similar failure.

The *Failure Response Cycle* also comprises the Failure Response process and the Accountability Achievement process. It begins when a failure has occurred or is predicted—in other words, when deviation from the allowable range of service levels is observed. Instructions for monitoring and recording system status and for recovery of the system from failures are denoted in D-Scripts.

D-Case is a structured notation based on the assurance cases[6] in the form of GSN[7], and it is used to describe the agreements reached by stakeholders concerning the requirements placed on the system and the methods to be used to satisfy them. The agreement description database, which we call D-ADD, stores D-Cases and D-Scripts on an ongoing basis to support accountability achievement.

When first looking at this dual-cycle structure, you may see it as a state transition chart—a token placed on the Ordinary Operation state, when an anomaly or failure is detected, moves to the Failure Response state, and then moves to either the Consensus Building state or the Accountability Achievement state, and so forth. This is, however, not correct: since the system must continue operating to the greatest extent possible in order to deliver services to users without interruption, all of the components of the dual-cycle structure must be realized as concurrent processes. That is, when an anomaly or failure is detected in the Ordinary Operation state, the Failure Response process is activated for this event, but the Ordinary Operation state keeps operating to the greatest extent possible. The Failure Response process activates the Accountability Achievement process after it has finished the corresponding responsive actions. It may activate the Consensus Building process, and so forth. The Failure Response process for the event can end after completion of its tasks.

Similarly, when a change in environments or objectives is recognized as requiring modification of the system, the Ordinary Operation state activates the Consensus Building process for this event, and it then activates the development process, and so forth. When two or more events have activated the same process in parallel or consecutively in an overlapped manner, these activated processes may need to collaborate for a consistent solution. Thus, the DEOS process is one of concurrent processes.

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

## IV. RETROSPECTIVE

The DEOS project began in 2006 under the support of the Japan Science and Technology Agency (JST). The goal of the project was to develop dependable operating systems for embedded systems. The members of the project realized that the embedded system we needed to address in reality was an application on a terminal operating in conjunction with a back-end server, not an isolated one. Thus, we developed a *dependable embedded OS* that provides enhanced hierarchical isolation of processes with monitoring and logging, taking server systems into consideration.

In the meantime, we began to doubt whether dependability could be assured with an OS alone and even whether absolute dependability was achievable. Following this, after many long discussions, the DEOS project members reached a common recognition of the need for a new definition of dependability for open systems. As a result, we proposed the definition of Open Systems Dependability and developed the DEOS process, which is iterative and comprises concurrent processes for achieving accountability based on stakeholders' agreements.

In parallel with these efforts, we developed an extension of the assurance case called the D-Case for use in consensus building regarding the requirements, specifications, and implementation of system operation. We also developed D-Script to provide scripting functionality for monitoring, recording, and control during system operation.

We began work aimed at international standardization, and as a result, The Open Group (TOG) industrial standards organization published the *Dependability through Assuredness$^{TM}$ (O-DA) Framework* in July 2013. In parallel with this, we started efforts to establish a *de-jure* international standard, and the International Electrotechnical Commission is in the process of standardizing IEC 62853, *Open Systems Dependability*. Furthermore, a user consortium—the Association of Dependability Engineering for Open Systems (DEOS Association) —was established in October 2013 in order to promote the application and evolution of DEOS.

## V. Prospects

As we already know from our work and daily lives, many of today's systems operate online for extended periods of time, accommodating change in their environments and objectives. They are composed of many subsystems while also connected to external systems in order to request tasks and receive information. In the advent of the IoT, everything is connected, meaning that the number of interconnected devices and servers has become uncountable and uncontrollable. Some of them are reliable, but some contain bugs, are anonymous and thus irresponsible, and can be counterfeit or malicious. Big data collected via the IoT may include inaccurate, false, and old information. AI delivers convenience in a great many situations, but we must consider the extent to which we can rely on the recommendations and predictions of AI systems, given that they are generated from the big data.

In such an era, it is clear that we cannot apply constructive approaches to systems beyond what we really know and control. We must admit that we are dealing with open systems, and only an iterative approach can be applied to the dependability of such systems.

Iterative approaches have been proposed and applied, typical examples being PDCA[8] and OODA[9]. PDCA consists of four steps—Plan, Do, Check (or Study), and Act—and it has been used to improve the quality of products and organizations. OODA comprises four concurrent processes—Observe, Orient, Decide, and Act. It was originally devised for combat pilots for making quick, pertinent decisions in order to defeat their enemies, and it was then applied to business.

The DEOS process, which shares the fundamental concept of the above-mentioned methods, has been developed for improving the dependability of software in a practical way. It has a dual-cycle with concurrent processes for achieving accountability based on the stakeholders' agreements.

The DEOS process has been applied, for example, to the development of robots and distributed energy systems, and further application is planned in the development of automotive software, including self-driving. Other application domains including intelligent transport systems, cloud servers, disaster management planning, medical systems, and smart factories are currently under consideration.

With publication of international standard IEC 62853 and increased recognition of today's software and other systems operating in open-system environments, application of the DEOS process is expected to become more widespread and common.

## VI. Conclusion

In this paper, I presented the concept of open systems and open systems dependability, summarized the DEOS process, presented a retrospective on the DEOS project, and extended prospects for the DEOS process. Although developed for the improvement of software dependability, the concept of open systems dependability and the DEOS process can be applied to various engineering domains and also to business and social domains. I believe that the concept of open systems dependability and the DEOS process will contribute to the dependability of such systems, and ultimately, to that of our society.

## Acknowledgements

I would like to thank all members of the DEOS project, the DEOS Association, and Sony Computer Science Laboratories, Inc. for their invaluable, continuous discussions on open systems, open systems dependability, the DEOS process, and the applications thereof. I also wish to thank the Japan Science and Technology Agency (JST) for supporting the DEOS project and the Information-technology Promotion Agency (IPA) for its support of the DEOS Association.

## References

[1] L. Bertalanffy, General System Theory: Foundations, Development, Applications, G. Graziller, New York, 1968.

[2] Karl Popper, The Logic of Science Discovery, Hutchinson & Co., 1959.

[3] M. Tokoro (ed), Open Systems Science – from Understanding Principles to Solving Problems, IOS Press, Amsterdam, 2010.

[4] M. Tokoro, Open Systems Science – A Challenge to Open Systems Problems, `http://cs-dc-15.org/papers/governance/open-systems-explo/open-systems-science-a-challenge-to-open-systems-problems/`, 2016.

[5] M. Tokoro (ed.), Open Systems Dependability – Dependability Engineering for Ever-Changing Systems, 2nd edition, CRC Press, 2015.

[6] R. Bloomfield, P. Bishop. "Safety and Assurance Cases: Past, Present and Possible Future — an Adelard Perspective", Proceedings of the Eighteenth Safety-Critical Systems Symposium, Bristol, UK, 9-11th February 2010, pp. 51-67.

[7] The GSN Working Group, "GSN Community Standard, Version 1", 2011.

[8] Please refer to Wikipedia.

[9] Please refer to Wikipedia.

## About the Author

Dr. Mario Tokoro is president of the Association of Dependability Engineering for Open Systems (DEOS Association) and also of the Institute for Open Systems Science. He is a former professor of computer science at Keio University, the founder of Sony Computer Science Laboratories, Inc., and former Senior Vice President and Chief Technology Officer of Sony Corporation. He is an expert in computer science and engineering, the philosophy of science and technology, and research management.