Software cybernetics: current state and problems

Evseeva Yulia Igorevna

Candidate of engineering science, Penza State University, 89374447008
shymoda@mail.ru

1 Introduction

Software cybernetics is a relatively young scientific direction focused on applying the methods of classical cybernetics to the tasks of software engineering. And although the basic principles of both disciplines are now well-developed and studied, in software cybernetics there is still no universally accepted view of the representation of software processes as cybernetic. Also, the boundaries of the field under consideration are rather blurred. The purpose of this article is to briefly review the main works in this field and on its basis to determine the structure, trunk directions and current achievements of program cybernetics.

As the main material for the study, the works of the world's leading scientists in the field of system analysis, software engineering and artificial intelligence were used.

The main results include: a review of existing works in the field of program cybernetics, a classification of its main tasks, problems and directions.

The review showed the relevance of program cybernetics as a scientific direction, within the framework of which it is possible to successfully solve a number of topical tasks for today, related to the creation and operation of complex self-adaptive and scalable software systems.

2 Theoretical foundations of software cybernetics

Software systems are complex in nature. The complexity of design, implementation and maintenance exponentially grows with system size increase. It is difficult, and often impossible, to list all the states and interactions that characterize modern software components.

In addition to complexity, software systems are often characterized by the need for modification in the process of operation. This can be caused both by changes in the system subject area, and by various hardware problems.

Research in the field of cybernetics makes it possible to control software systems complexity and to provide them with adaptive properties, thereby making them more flexible and effective.

The term “software cybernetics” was first used by K. Cai in 2002 [1]. Originally it designated an attempt to apply the methods of classical cybernetics and control theory to software systems. Since then, this new field of scientific knowledge has undergone significant changes and currently includes not only the implementation of basic cy-

bernetic principles with respect to software, but also original concepts and methodologies. The latter are elements of so-called "new" cybernetics. This article includes a full review of both classical and innovative approaches in software cybernetics.

In K. Cai's work [1] software cybernetics is presented in its simplest form – as a way of organizing interaction between the software system and the control unit. According to another, later work of this author [2], software cybernetics should be aimed at solving the following problems:

1. formalization and quantification of feedback in software processes and systems;
2. adaptation of control theory principles and concepts to software processes and systems;
3. application of software engineering principles to control systems and processes;
4. integration of software engineering and management engineering.

According to [3], software cybernetics, like the classical one, can be divided into first and second order cybernetics. First-order software cybernetics is focused on modeling software systems as being managed on the basis of feedbacks. The most common model for formalizing control processes in this case is the finite automaton. Second-order software cybernetics considers more complex systems, the components of which are software and software development and maintenance processes, as well as development teams. All elements of such systems influence each other.

At present, software cybernetics includes 2 main groups of methods for researching and modeling software systems: model-oriented and logic-oriented.

The first group includes various mathematical methods based on the use of finite automata and Markov chains, as well as models of linear dynamical systems. For example, [4] describes the process of testing software as a linear dynamic system. P. Wang and K. Cai have developed algorithms that allow the transformation of extended finite automata, used to describe processes in specification and description language (SDL) [5], into the model of discrete-event systems control. Their research has shown that an extended finite automaton can be used to describe a closed-loop control system. In turn, D. Lorenzoli proposed a method of automatic model generation for determining the software systems behavior based on finite mathematical automata [6].

Progress in artificial intelligence contributes to the development of the second group of methods. Software engineering has become one of the main areas of machine learning algorithms application. K. Yang used fuzzy logic methods to create software tools that allow developing self-adapting software systems [7].

The ultimate goal of the work was to improve mission-critical software performance and fault tolerance. Z. Ding proposed the concept of adaptive control system based on fuzzy rules [8]. K. Park and K. Yeom used the idea of feedback to implement SWRL rules validating method (Semantic web rule language). According to the proposed approach, each SWRL rule under consideration is a control object, and rules correctness is checked by using a specific control unit [9].

The introduction of artificial intelligence technologies will ultimately bring software cybernetics to a new qualitative level – the level of the third order cybernetics.

3 Practical applications of software cybernetics

To date, first-order software cybernetics has found application in the following areas of software engineering:

1. development of specifications and requirements for software;
2. designing the software systems architecture;
3. implementation (programming);
4. testing;
5. software reconstruction and development;
6. project management;
7. information security.

Application areas of the second order software cybernetics include:

1. cloud computing;
2. service-oriented interactions;
3. agent technologies.

Also, program cybernetics is closely integrated with the following technological trends underlying the cyber physical systems:

1. network systems;
2. Internet of things;
3. big data;
4. creative calculations.

Let us consider the latest research in the field of software engineering.

The work of L. Liu, published in 2015 [10] is of the greatest interest in specifications development area. In this work the author proposes a mechanism for identifying requirements for software based on user behavioral data analysis.

The process of identifying requirements is presented as a feedback management system. Under this approach, requirements' identification is reduced to continuous optimization of user behavior patterns.

The software development community has for many years been occupied with the problem of creating adaptive software systems. Developers have proposed a lot of approaches to such systems construction, many of which are based on the principles of machine learning and control theory. Particular attention was paid to management theory, since it provides general methodology for creating adaptive systems. Based on the experience accumulated during adaptive software systems development, in 2005 the concept of autonomous computing was formulated [11]. Autonomous computing is an intelligent approach to building self-managed software and hardware systems capable of maintaining a stable state of the computing environment with minimal human participation. The software of such complexes is capable of changing its own structure and behavior in the process of execution in order to ensure stable operation of the system under constantly changing requirements and environmental conditions.

In software engineering the software reconstruction is an approach that helps to prevent performance degradation and other problems associated with software obsolescence. Existing reconstruction methods can be divided into 2 groups: model-based methods and measurement-based methods [12]. Most of the efforts in this direction are focused on predicting possible program outage and planning an optimal update strategy [13]. R. Agepati offers a model of software reconstruction, based on the feedback mechanism [14]. The proposed reconstruction model includes the description of software obsolescence process, a set of update actions and an obsolescence detection strategy. The optimal strategy for system reconstructing is based on the principle of minimum cost and is implemented using the Markov decision-making process.

Cloud technologies have become the dominant computing environment of the current decade, thanks to their ability to quickly provide computing resources with minimal operating costs. Since systems based on cloud technologies are complex, large-scale, distributed and heterogeneous, managing their resources is not an easy task. In order to solve this problem P. Meyer proposes the concept of an autonomous cloud [15] – a cloud using the volunteer computing principle and built on the basis of a peer-to-peer network. Volunteer computing refers to distributed computing using voluntarily provided computational resources. Applications execution management is based on the ideas of multicast reflection and self-adaptation. However, at the moment many aspects of such concept implementation require research.

In addition to cloud computing, in recent years strong interest has been evoked by systems based on service-oriented interactions. Service-oriented computing is a computational paradigm that uses services as main resources for creating software systems. Service-oriented architecture is a composition of various software services, each of which is responsible for solving its own autonomous task. K. Liu designed a method for solving the adaptation problems in service-oriented systems, based on control theory ideas [16]. At present, there are relatively few works devoted to the integration of software cybernetics and agent technologies, despite the fact that multi-agent systems nature seems very close to this direction. From the cybernetic point of view, the agent system is a system of individuals who, interacting and exchanging information, seek to solve some common problem. K. Sim offers an agent-based paradigm for managing resources in cloud systems [17]. Innovative elements of his work include: agent-based search engine for detecting cloud services, agent-based resource matching mechanism and problem-solving technique that occurs when integrating cloud services.

K. Revidran applies software cybernetics to control the complex network systems behavior [18]. The proposed approach assumes the use of model-oriented software engineering techniques for assessing the quality of network system adaptation in uncontrolled environment. T. Choi [19] discusses the possibility of using cybernetic principles in collection and analysis of big data. Also, from the point of view of optimal controllability, decision-making and adaptability, the problems of integrating information and technical systems are relevant within the methodology of the Internet of things. The paradigm of the Internet of things is based on widespread introduction of intellectual technology into the society, which gives rise to a number of new prob-

lems for software engineering: the organization of intellectual objects interaction on a large scale, data collection and processing, etc.

4 Conclusion

There is no doubt that modern society lives in the era of ubiquitous software proliferation. Modern products and services increasingly include, as a component, software systems or are under their control. Environment unpredictability, rapidly changing demands and operating conditions require the creation of new ways of software development. The software should become more intelligent, self-organizing, resource-efficient and reliable. Modern software cybernetics poses new challenges for researchers and offers developers new opportunities related to the creation, operation and development of software systems.

References

1. Cai, K. Y. Optimal software testing and adaptive software testing in the context of software cybernetics. // Information and Software Technology.. – Hong Kong: 2002. – P. 841-855.
2. Cai, K. Y., Cangussu, J. W., DeCarlo, R. A., Mathur, A. P. An overview of software cybernetics // IEEE International Workshop on Software Technology and Engineering Practice. – Washington: IEEE Computer Society, 2003. – P. 77-86.
3. Kenett, R. S. Future directions of software cybernetics: A position paper. // 35th IEEE Annual Computer Software and Applications Conference Workshops. – Washington: IEEE Computer Society, 2003. – P. 43-44.
4. Cai, K. Y., Jiang, C. H., Hu, H., Bai, C. G. An experimental study of adaptive testing for software reliability assessment // J. Syst. Software.. – Norwell: Kluwer Academic Publishers, 2008. – P. 1406-1429.
5. Wang P., Cai, K. Y. Representing extended finite state machines for SDL by a novel control model of discrete event systems // Sixth IEEE International Conference on Quality Software (QSIC 2006). – Washington: Ieee Computer Society, 2006. – P. 159-166.
6. Lorenzoli, D., Mariani, L., Pezzè, M. Automatic generation of software behavioural models // 30th international ACM conference on Software engineering. – New York: ACM , 2008. – P. 501-510.
7. Yang, Q., Lü, J., Xing, J., Tao, X., Hu, H., Zou, Y. Fuzzy control-based software self-adaptation: A case study in mission critical systems // IEEE 35th Annual Computer Software and Applications Conference Workshops (COMPSACW). – Washington: IEEE Computer Society, 2011. – P. 13-18.
8. Ding, Z., Wei, Z., Chen, H. A software cybernetics approach to self-tuning performance of on-line transaction processing systems // J. Syst. Software.. – Amsterdam: Elsevier, 2016. – P. 13-18.
9. Park, K., Yeom, K. A feedback-based approach to validate SWRL rules for developing situation-aware software // 37th Annual Computer Software and Applications Conference Workshops (COMPSACW). – Washungton: IEEE Computer Society, 2013. – C. 41-46.
10. Liu, L., Zhou, Q., Liu, J., Cao, Z. Requirements cybernetics: elicitation based on user behavioural data // J. Syst. Software. – Amsterdam: Elsevier, 2016.

11. Ahuja, K., Dangey, H. Autonomic Computing: An emerging perspective and issues // IEEE International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT 2014). – Washington: IEEE Computer Society, 2014. – P. 471-475.
12. Cotroneo, D., Natella, R., Pietrantuono, R., Russo, S. Software aging and rejuvenation: Where we are and where we are going // IEEE Third International Workshop on Software Aging and Rejuvenation (WoSAR 2011). – Washington: IEEE Computer Society, 2011. – P. 1-6.
13. Okamura, H., Dohi, T. Application of reinforcement learning to software rejuvenation // 10th International Symposium on Autonomous Decentralized Systems (ISADS). – Washington: IEEE Computer Society, 2011. – P. 647-652.
14. Agepati, R., Gundala, N., Amari, S. V. Optimal software rejuvenation policies // IEEE conference on Reliability and Maintainability Symposium (RAMS 2013). – Washington: IEEE Computer Society, 2013. – P. 1-7.
15. Mayer, P., Klarl, A., Hennicker, R. The autonomic cloud: a vision of voluntary, peer-2-peer cloud computing // 7th IEEE International Conference on Self-Adaptation and Self-Organizing Systems Workshops (SASOW). – Washington: IEEE Computer Society, 2013. – P. 89-94.
16. Liu, C., Jiang, C., Hu, H., Cai, K. Y., Huang, D., Yau, S. S. control-based approach to balance services performance and security for adaptive service based systems (ASBS) // 33rd Annual IEEE International Computer Software and Applications Conference (COMPSAC'09). – Washington: IEEE Computer Society, 2009. – P. 473-478.
17. Sim, K. M. Agent-based cloud computing // IEEE Transactions on Services Computing. – Washington: IEEE Computer Society, 2012. – P. 564-567.
18. Ravindran, K., Rabby, M. Software cybernetics to infuse adaptation intelligence in networked systems // Fourth IEEE International Conference on the Network of the Future (NOF). – Washington: IEEE Computer Society, 2013. – P. 1-6.
19. Choi, T., Chan, H., Yue, X. Recent development in big data analytics for business operations and risk management // IEEE Transactions on Cybernetics. – Washington: IEEE Computer Society, 2016. – P. 1-12.