

# Software Cybernetics in the Age of Cloud Computing

## -- Challenges and Opportunities

Hong Zhu

Department of Computing and Communication Technologies, Oxford Brookes University  
Oxford OX33 1HX, UK, Email: [hzhu@brookes.ac.uk](mailto:hzhu@brookes.ac.uk)

As "*the scientific study of control and communication in the animal and the machine*" [1], the past six decades have seen a rapid development of cybernetics and its applications to a wide range of scientific, technology and engineering disciplines, including biology, psychology, sociology, art, management, engineering and computer science. IWSC 2012 marks the 9<sup>th</sup> annual workshops devoted to software cybernetics, which is perhaps the unique international academic forum on this topic. In this talk, I will briefly review some examples of research in the area of software cybernetics, for example:

- *Search-based software engineering*—an increasing popular research method on software engineering that regards many software engineering problems as optimization problems and explicitly utilizes meta-heuristics and "*close control loop*" to search for the best solutions [2];
- *Software engineering based on the wisdom of crowds*—an novel approach to develop quality software based on swarm intelligence [3];
- *Scenarios calculus* – my own attempt to formalize and reasoning about emergent behaviors in multi-agent software systems as a part of a new paradigm of agent-oriented software development methodology [4] [5]; etc.

Software cybernetics, as a subdivision of cybernetics in the domain of software engineering, has manifested itself a number of distinctive features. In this subdivision, on the first order level, the object or system under observation and control are software and the controllers are also software. On the second order level, the software systems and their developers and users form a more complicated system. Thus, the development processes and maintenance and evolution of software systems are observed and controlled, even optimized and automated, for example, through software tools and environments. In this talk, I will further examine the key features of software that distinguishes software systems from other types of systems for which subdivisions of cybernetics have successfully developed and flourished with successful applications. I will refer to Brooks' four essences of software engineering, i.e. complexity, conformity, changeability and invisibility [6] to discuss how these features manifest themselves to impose challenges to software engineering in the age of cloud computing and service-oriented computing. In particular,

- *complexity* of software systems comes to a higher level, where services can be dynamically discovered and composed at runtime into composite services.
- *Conformity* is not limited to platform and environment, but also to the semantics of services when they are dynamically searched and composed.
- *Changeability* is not just because users' requirements and platform changes, but also the third party services changes, even the ontology in which semantics of services are defined and the human society is changing.
- *Invisibility* becomes an even harder problem because the documents, even the source code, of the software from third party services are not available.

These challenges also offer opportunities to the research on cybernetics to develop a flourishing subdivision. For example, a cloud is a complicated system that consists of infrastructure-as-a-service, platform-as-a-service and software-as-a-service. It must be controlled to achieve the quality of services in many different measures to satisfy the service level agreements for various customers. This is an ever complex monitoring and control problem.

In addition, software cybernetics may shed new light to the following problems in software engineering that are becoming more and more important in the age of cloud computing.

- *Emergent behavior in service oriented architecture.* Regarding services as autonomous active entities, their communication and collaboration mechanisms are of particular importance for the emergent behavior of such software systems. How to specify, design, implement verify, validate and test such systems are still unknown to the service engineering. The theories of complexity in cybernetics could shed a new light to this problem.
- *Software evolution in the cloud.* Much research on software evolution have been done in the past years, but the impact of software changes in cloud computing is much severer and even before. Could cybernetics provide new thought to this?
- *Self-adaptive architecture.* As software as a service becomes more and more popular, it is required to be adapted into difference environment and meet the needs of different types of users. Self-adaptiveness will inevitably become a necessity rather than an optional feature. Could cybernetics provide a solid

foundation for this?

- *Software metrics.* Measurement is the basis of control. A large number of different software metrics have been proposed in the past decades and much research effort has been reported in the literature. From cybernetics point of view, a question that remains open is that whether these metrics can be used in the control of software and software development and evolution process?

## References

- [1] N. Wiener, *Cybernetics: Communication and Control in the Animal and the Machine*, Cambridge: MIT Press, 1948.
- [2] M. Harman, P. McMinn, J. Souza and S. Yoo, "Search-based software engineering: Techniques, taxonomy, tutorial," in *Empirical software engineering and verification: LASER 2009-2010*, vol. LNCS 7007, B. M. a. M. Nordio, Ed., Springer, 2012, pp. 1-59.
- [3] H. Wang, "Trustie: Trustworthy Software Development Environment based on the Wisdom of Crowds, Keynote Speech at SOSE 2011," Dec. 2011. [Online]. Available: <http://tech.brookes.ac.uk/sose2011/SOSE-KeynoteHMWang.pdf>. [Accessed 6 July 2012].
- [4] H. Zhu, F. Wang and S. Wang, "On the Convergence of Autonomous Agent Communities," *Multi-Agent and Grid Systems: An International Journal*, vol. 6, no. 4, pp. 315-352, Dec. 2010.
- [5] H. Zhu, "Formal Reasoning about emergent behaviour in MAS," in *Proceedings of SEKE '05*, Taipei, 2005.
- [6] P. F. Brookes, "No Silver Bullet: Essence and Accidents of Software Engineering," in *Information processing*, H. Kugler, Ed., North-Holland, 1986.