SOFTWARE CYBERNETICS

October 7, 2009

The Software Testing and Software Cybernetics Convergence: *What, Why, How and Who*

A white paper by R. Kenett[1], D. Almog[2], Joao Cangussu[3] and Xiaoying Bai[4,*]

What

This white paper is prepared, as a discussion platform, for the Software Cybernetics Special Interest Group (SC-SIG): http://www.linkedin.com/e/vgh/2189748.

The SC-SIG combines members from a variety of background and disciplines. It includes academics, practitioners, experts in software engineering, technology developers, control engineers, IT managers, and statisticians.

The discussion we hope this paper will generate are welcome to be posted on the LinkedIn site. Eventually, we plan to collect these ideas, organize them and present them at the 2010 International Workshop on Software Cybernetics (IWSC).

The idea is to be clear, realistic, directly to the point and sometimes blunt and/or provocative. Our goal is to generate new knowledge and understanding that can have concrete impact on industry. Though the scope of software cybernetics is much broader, we focus here on the interplay between software testing technology and control concepts.

[1] KPA Ltd., Israel, University of Torino, Italy and NYU-Poly Ctr for Risk Engineering, USA, ron@kpa.co.il

[2] Ben Gurion University, Israel, almog.dani@gmail.com

[3] University of Texas-Dallas, USA, cangussu@utdallas.edu

[4] Tsinghua University, China, baixy@tsinghua.edu.cn

[*] coordinating partner

Furthermore, we analyze software cybernetics with respect to Services Computing Architecture, SOA, and SaaS concepts. There are at last three gaps between academic research and industry regarding these systems:

Gap 1: Testing Platforms
This gap is pointing at a required transition from human-centric, manual approach to automatic testing. Here we need at least a test platform that can automatic load and execute the test scripts based on the requirements and/or code - **very few such platforms exist**.

Gap 2: Open Systems
Here we need to move from a closed, product-oriented test environment, to an open, service-based, collaborative test culture supported by advanced testing systems - **very few such systems exist**.

Gap 3: Intelligent Testing
This is the more challenging gap. Moving from "blind" testing, to semantic-enabled intelligent testing reflecting actual and intended usage of the system under test.

Software Testing is a fundamental activity within the software development process that exposes software product behavior and faults, impacting the quality of the final product. It provides an opportunity for bugs to be discovered and corrected prior to the software deployment. Clearly, testing should be a reflection of user requirements. Validation testing (sometimes called qualification testing or acceptance testing) is based on requirements and is often done manually. Automated test scripts, when performed, are also developed manually and are based on the testers concept of what that requirement is, and the perception may be flawed. What is needed is better user-oriented and risk based testing.

Software cybernetics explores the interplay between software/systems behavior, and control. The fundamental question of interest is how to adapt software behavior, software processes, or software systems to meet basic and expanded objectives in the presence of a changing environment. Changes are due to disturbances, faults, or revised requirements.

The initiative to push forward a convergence between Software Testing and Software Cybernetics, is designed to help meet challenges of new computer systems such as Service Oriented Architecture and Services Computing applications. The focus here is on expanding the "test broker" concept with further testing methodology and technology.

Why

Modern systems such as Services Computing and Web Services require sophisticated control strategies to operate. Runtime monitoring is necessary for continuous quality assurance of Web Services. Sensors are critical in runtime monitoring to capture the data and detect anomalies. However, sensors in current monitoring systems are usually manually instrumented or hard-coded in the program and it is expensive to implement, and inflexible to change at runtime. Moreover, testing and evaluation is necessary in the lifecycle of service-based software development. In that context, users can choose between alternative services and identify the one that best satisfy their needs of functionalities and quality. Dynamic testing provides a mechanism generating information that allows users to make an adequate choice. The convergence of Testing technology and Software cybernetics is an approach that can yield effective and efficient solutions in new services computing applications.

How

Software Cybernetics is an emerging and interdisciplinary area that addresses issues and questions such as 1) Formalization and quantification of feedback and self-adaptive control mechanisms in software; 2) Adaptation of control theory principles to software processes and systems; 3) Application of software engineering principles and theories to control systems and processes and 4) Integration of the theories of software engineering and control engineering. By combining Software Cybernetics and Testing Technology we can derive a new paradigm for "Test Brokers" that parallel the development and operation of modern computer applications. The implementation of dynamic testing requires complex technology deployment. That technology consists of several components including, test generation, test result assessment, data contexts and operational profiles.

The setting up of a data context is particularly important. In this task, a match between the service contract and the generation of test data is achieved, as context for the testing activity. This has been one of the barriers to effective testing of SOA systems. Another technological requirement is the availability of an effective test oracle that provides for an interpretation of the test outcomes. Such an oracle requires **measurement**s and **rules**

Measurement are quantitative evaluation of the current state of the software system against the goal. The rules build up the roadmap from current state to the goal, based on the measurement .

For example, in Risk-Based Adaptive Group Testing of Web Services (Bai and Kenett, COMPSAC/IWSC 2009) the goals are: 1) minimum number of test cases; 2) maximum discovered defects; 3) maximum covered software elements (ontologies and the data defined by the ontology). The measurements track the state of the test cases and ontologies, such as the potency of each test case (the ratio between the number defects discovered and the number of total runs), and the failure probability of each ontology given the current defects detected and the relationships between the ontologies. The rules tell us: which should be the next test case to select so that we have the highest probability to achieve the goal .

A key to the process is to enable continuous updates of the measurements and adaptation of the software behavior based on the rules .

Moreover, as software engineering is a layered technology, from process to activities to technologies and tools, we may apply the generic framework to various areas of software engineering practices, such as dynamic architecture, adaptive testing and software modernization (evolution).

Who

Leading team: Ron Kenett Dani Almog, Joao Cangussu and Xiaoying Bai,

There are several groups and organizations that participate in this initiative. They include:

- The Software Cybernetics - Special Interest Group: www.linkedin.com/e/vgh/2189748
- The International Workshop on Software Cybernetics (IWSC) which has been traditionally collocated with COMPSAC: www.utdallas.edu/~cangussu/site/IWSC09
- KPA Ltd., Israel
- University of Turin, Italy
- NYU-Poly Center for Risk Engineering, USA
- Ben Gurion University, Israel
- University of Texas-Dallas, USA
- Tshingua University, China

You are welcome to join the group and contribute to the discussions and ideas presented in the Software Cybernetics- Special Interest Group, the annual conferences and various email exchanges.