Cardoso, J., "Evaluating the Process Control-flow Complexity Measure", 2005 IEEE International Conference on Web Services (ICWS 2005), July 12-15, 2005, Orlando, Florida, USA. pp. 803-804, ISBN: 0-7695-2409-5.

Evaluating the Process Control-flow Complexity Measure

Jorge Cardoso
Department of Mathematics and Engineering
University of Madeira
9050-390 Funchal, Portugal
jcardoso@uma.pt

Abstract

Process measurement is the task of empirically and objectively assigning numbers to the attributes of processes in such a way as to describe them. We define process complexity as the degree to which a process is difficult to analyze, understand or explain. One way to analyze a process' complexity is to use a process control-flow complexity measure. This measure analyzes the control-flow of processes and can be applied to both Web processes and workflows. In this paper, we discuss how to evaluate the control-flow complexity measure to ensure that it can be qualify as a good and comprehensive one.

1. Introduction

Business Process Management Systems (BPMS) provide a fundamental infrastructure to define and manage business processes. The emergence of processes that span both between and within enterprises [1] have an inherent complexity. Expected benefits of BMPS solutions indicate that 38% will happen in enterprise-wide redesign of business process (source Delphi Group 2002).

One of the major problems of large processes, such as cross-organizational processes, is their high complexity. Unfortunately, no methods have yet been developed to analyze and decrease the degree of complexity. Therefore, a lot of time is spent reading and understanding processes in order to remove faults or adapt the process to changed requirements.

For example, mission-critical healthcare processes are often complex, large-scale, and QoS-based [2]. These processes are also very dynamic since changes in healthcare treatments, drugs, and protocols may invalidate running instances, requiring reparative actions. Furthermore, large-scale processes often span multiple healthcare organizations and run over long periods of time. These requirements make it necessary to determine specific characteristics of processes with engineering methods to guarantee a correct execution. Important characteristics to analyze include the estimation of complexity, process size, effort of testing, effort of maintenance, understandability, and quality to enable the correction of problems before any drastic consequences occur.

2. Process Measurements

Process measurement is a field that presents a set of approaches to the quantification of specific characteristics of processes. This field is still in its infancy and much work has yet to be undertaken.

Process measures can and should be applied in every phase of the process development life-cycle, including the analysis, design, implementation, testing, and maintenance [3] phases. The idea of applying process measurement in the early phases of the process life-cycle is to improve process development in the process design phase by a feedback controlled by process measures in order to get a better implementation of processes in the coding phase and a less complicated and less expensive maintenance.

3. Control-flow Complexity

In [4] we have proposed a complexity measurement, called control-flow complexity (CFC), to be used during the design of processes. This design-time metric can be used to evaluate the difficulty of producing a designed process before the actual implementation exists.

Using control-flow complexity measurements we can improve the comprehensibility of processes. Processes can be reengineered to reduce the complexity of related activities. One key to the reengineering is the availability of a metric that characterizes complexity and provides guidance for restructuring processes.

The use of CFC analysis will aid in constructing and deploying Web processes and workflows that are more reliable and robust. The following benefits can be obtained from the use of complexity analysis:

- Quality assessment. Processes' quality is most effectively measured by objective and quantifiable metrics. Complexity analysis allows calculating insightful metrics and thereby identifying complex and error prone processes.
- Maintenance analysis. The complexity of processes tends to increase as they are maintained, and over a period of time. By measuring the complexity before and after a proposed change, we can minimize the risk of the change.
- Reengineering. Complexity analysis provides knowledge of the structure of processes. Reengineering can benefit from proper application of complexity analysis reducing the complexity of processes.

4. Software and Process Metrics

Research in software engineering has produced various measurements for software. Among others are the Halstead's measure, McCabe's measure, the COCOMO model and the Function-Point method. Zuse [5] has found hundreds of different software metrics proposed and described for software measurement.

While a significant amount of research on the complexity of software programs has been done in the area of software engineering, the work found in the literature on complexity analysis for Web processes, workflows, and processes in general is almost inexistent.

5. Evaluating CFC

As we have stated previously, in the area of software measurement, the methods and theory developed have had a reduced industrial acceptance. According to some research, another reason is that there is a lack of serious validation of proposed metrics; and thus, a lack of confidence in the measurements.

To overcome this difficulty we believe that our CFC metric can be evaluated according to Weyuker's [6] properties, a method used in the field of software engineering. These properties have been seriously discussed in literature. Although these properties have been also criticized [7], currently they are still subject to refinement. Weyuker's properties are a widely known formal analytical approach and were therefore chosen for our analysis since they do provide a basis for some validation of complexity metrics.

Our CFC measure happens to fully satisfy seven of the above properties and partially satisfies one property it can be considered to have passed a significant part of the theoretically validation process.

6. Conclusions and future work

In our previous research we have proposed a process control-flow complexity (CFC) metric to assert the difficulty of producing Web process and workflow designs before an actual implementation exists. To evaluate the proposed CFC metric, we borrow theory from the software engineering field. Namely, we use Weyuker's properties to classify our CFC measure to determine if it can be categorized as a good, structured, and comprehensive one. The results of the evaluation were positive; classifying the CFC metric has a good and structured metric. Our next step will be to evaluate our metric using empirical testing involving human subjects and their perception of process complexity.

References

1. Sheth, A.P., W.v.d. Aalst, and I.B. Arpinar, Processes Driving the Networked Economy. IEEE Concurrency, 1999. 7(3): p. 18-31.

2. Cardoso, J., A. Sheth, and J. Miller. Workflow Quality of Service. in International Conference on Enterprise Integration and Modeling Technology and International Enterprise Modeling Conference (ICEIMT/IEMC'02). 2002. Valencia, Spain: Kluwer Publishers.

3. Cardoso, J. Building workflows definitions based on business cases. in First International Workshop on Computer Supported Activity Coordination (CSAC-2004), ICEIS 2004, 6th International Conference on Enterprise Information Systems. 2004. Porto, Portugal.

4. Cardoso, J., How to Measure the Control-flow Complexity of Web Processes and Workflows. 2004, Department of Mathematics and Engineering, University of Madeira: Funchal, Portugal. p. 15.

5. Zuse, H., A Framework of Software Measurement. 1997, Berlin: Walter de Gruyter Inc.

6. Weyuker, E.J., Evaluating software complexity measures. IEEE Transactions on Software Eng., 1998. 14(9): p. 1357-1365.

7. Cherniavsky, J.C. and C.H. Smith, On Weyuker's Axioms for Software Complexity Measures. IEEE Transactions on Software Engineering, 1991. 17(6): p. 636-638.