Cardoso, J., "About the Complexity of Teamwork and Collaboration Processes", IEEE 2005 International Symposium on Applications and the Internet (SAINT 2005), Workshop - Teamware: supporting scalable virtual teams in multi-organizational settings, Eds: Wojciech Cellary and Hiroshi Esaki, Trento, Italy, Jan 31 - Feb 4, 2005, IEEE Computer Society, ISBN: 0-7695-2050-2, pp. 218-221.

About the Complexity of Teamwork and Collaboration Processes

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

Abstract

Organizations across the globe are increasingly using teams to accomplish significant work and projects. Much of this work is also accomplished using technology tools to support their communication and collaborative efforts. As companies become increasingly multinational and distributed geographically, the formation of virtual teams has become a common practice. Workflow management systems are a specific type of systems that can be used to capture collaboration and group works processes and thus supports the creation of teamwork and enable collaboration. In some cases, collaboration and group work processes can become highly complex.

High complexity in a process may result in bad understandability and more errors, defects, and exceptions leading processes to need more time to develop, test, and maintain. Therefore, excessive complexity should be avoided.

The major goal of this paper is to discuss the need and requirements for the development of a measure to analyze the complexity of processes.

1. Introduction

The emergence of e-commerce has changed the foundations of business, forcing managers to rethink their strategies. Organizations are increasingly faced with the challenge of managing e-commerce applications, e-business systems, Web processes, and workflows.

Workflow Management Systems (WfMS) offer an environment to developers for the creation of processes to model virtual teams and collaborative work. These systems allow setting up the basic definitions for processes, the roles within an organization and the resources that are required to carry out a process. These systems can effectively carry out collaboration management, coordinating the activities of the participating applications and humans.

The emergence of processes that span both between enterprises and within the enterprise [1] have an inherent complexity. This complexity requires suitable methods to reengineer processes that are excessively complex, hard to understand and maintain. Studies indicate that 38% of process management solutions will be applied to redesign enterprise-wide processes (source Delphi Group 2002).

In a competitive e-commerce and e-business market, organizations want the Web processes and workflows that manage virtual teams to be simple, modular, easy to understand, easy to maintain and easy to re-engineer.

To achieve an effective management, one fundamental area of research that needs to be explored is the complexity analysis of processes. Process complexity can be viewed as a component of a QoS model for processes [2], since complex processes are more prone to errors. For example, in software engineering it has been found that program modules with high complexity indices have a higher frequency of failures.

We define process complexity as the degree to which teamwork and collaborative processes are difficult to analyze, understand or explain.

In our work we discuss the requirements of complexity metrics to be used during the development and maintenance of processes to obtain processes with a better quality and maintainability. The metric is used at design-time to evaluate the difficulty of producing a process before its implementation exists. When complexity analysis becomes part of the process development cycle, it has a considerable influence on the design phase of development, leading to further optimized processes.

2. Process Measurement

Process measurement is an activity assigning a number or a symbol to a process in order to characterize an attribute of the process according to given rules.

In the area of software measurement, many of the methods and theories developed to measure software complexity have had a reduced industrial acceptance. One of the reasons is that managers do not know what to do with the measurements. To surpass this difficulty we will discuss the issues, the requirements, and objectives of measures for the analysis of processes' complexity.

2.1. Entities, Attributes, and Metrics

The first step for the definition of any measurement is to give a precise definition for the concept intended to measure. If there is no clear statement of the concept to measure, it does not make much sense to talk about measurement.

We define three elements that are fundamental for the definition of any measurement: *entity*, *attribute*, and *metric*. The entities involved in our measurements are processes. A process can be measured according to different attributes (or dimensions [2]). The purpose is to provide a quantitative assessment of the extent to which the process possesses certain attributes. As an example, Table 1 illustrates various attributes that are of interest to be measured. The attributes/dimensions marked with the symbol '*' have already received some attention from researchers [2]. In our approach, the attribute that we will target and study is the complexity associated with a process.

Table 1. Process attributes/dimensions of interest to be measured

Attribute	Purpose
Complexity	How easy is the process to understand?
Functionality	Does the process satisfy user needs?
Cost*	What is the cost associated with the execution of a process?
Duration/Time*	How much time a process takes to execute?
Reliability*	How often does the process fail?
Usability	How easy is the process to use?
Efficiency	How good is the performance of the process?
Maintainability	How easy is the process to repair?

There is no single metric that can be used to measure the attributes of processes. Different metrics are needed for estimating a process complexity, functionality, readability, efficiency, maintainability, etc.

Several metrics can be defined for measuring the complexity of processes, namely:

- Control-flow complexity: The control-flow behavior of a process is affected by constructs such as splits and joins. Splits allow defining the possible control paths that exist through the process. Joins have a different role; they express the type of synchronization that should be made at a specific point in the process. A control-flow complexity model need to take into account the existence of XOR-split/join, OR-split/join, and AND-split/join constructs. The measure must be based on the relationships between mental discriminations needed to understand a split/join construct and its effects.
- Data-flow complexity: The complexity of a process increases with the complexity of data structures, the number of formal parameters of activities, and the data mapping between activities in a process. A data-flow complexity model needs to take into account the number and the complexity of the data structures that are transmitted between activities.
- Resources complexity: In a process, activities are carried out by means of resources (for example, a computer, a clerk or a DNA sequencing machine). The basic characteristic of a resource is that it has a certain capacity and is able to carry out particular activities. Resources can be classified according to their position in the organization or based on functional characteristics. Resources can be structured into groups and may belong to more than one group. A resources complexity metric needs to take into account all these aspects.

2.2. Definition of Process Complexity

Several definitions have been given to describe the meaning of software complexity [3]. After analyzing the characteristics and specific aspects of processes, we believe that the definition that is better suited to describe processes complexity can be derived from [4]. Therefore, we define process complexity as *the degree to which a process is difficult to analyze, understand or explain. It may be characterized by the number and intricacy of activity interfaces, transitions, conditional and parallel branches, the existence of loops, roles, activity categories, the types of data structures, and other process characteristics.*

2.3. Requirements

The development of a model and theory to calculate the complexity associated with a process need to conform to a set of basic but important properties. The metric should be easy to learn, computable, consistent and objective. Additionally, the following properties are also highly desirable:

- Simplicity. The metric should be easily understood by its end users, i.e., process analysts and designers.
- Consistency. The metric should always yield the same value when two independent users apply the measurement to the same process, i.e., they should arrive at the same result.
- Automation. It must be possible to automate the measurement of processes.
- Measures must be additive. If two independent processes are put into sequence then the total complexity of the combined structures is at least the sum of the complexities of the independent processes.
- Measures must be interoperable. Due to the large number of existing specification languages, both in the academia and the industry, the measurements should be independent of the process specification

language. A particular complexity value should mean the same thing whether it was calculated from a process written in BPEL, WSFL, BPML, YAWL, or some other specification language. The objective is to be able to set complexity standards and interpret the resultant numbers uniformly across specification languages.

These properties need to be taken into account when developing metrics to compute the complexity of processes.

2.4. Benefits of complexity analysis

Analyzing the complexity at all stages of process design and development helps avoid the drawbacks associated with high complexity processes. Currently, organizations have not implemented complexity limits as part of their process management projects. As a result, it may happen that simple processes to be designed in a complex way.

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

- Quality assessment. Process 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 (Figure 1). 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 the proper application of complexity analysis by reducing the complexity of processes.
- Dynamic behavior. Processes are not static applications. They are constantly undergoing revisions, adaptations, changes, and modifications to meet end users needs. The complexity of these processes and their continuous evolvement makes it very difficult to assure their stability and reliability. In-depth analysis is required for fixing defects in portions of processes having high complexity (Figure 1).

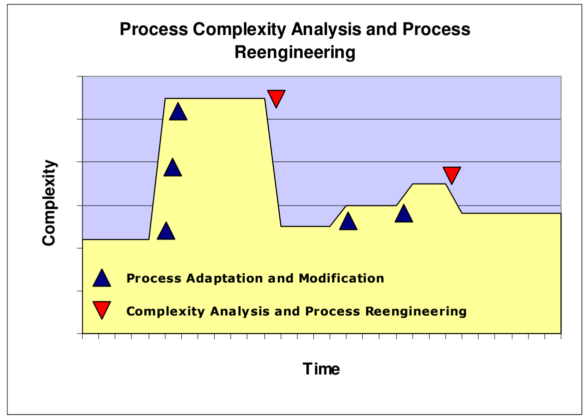

Figure 1. Process complexity analysis and process reengineering

2.5. End users

The complexity measurement enables process managers and administrator to calculate the complexity of process templates generated by others. Process designers can analyze the complexity of a particular process in development. Process consultants can contribute with new process components, needing methods to analyze the complexity of the proposed solutions. End-users can inquire about the complexity of processes before starting process instances.

2.6. Using complexity measurements

Let us consider two process applications: the eligibility referral application and the head ultrasound pathway. The eligibility referral application (Figure 3) was developed for the Connecticut Healthcare Research and Education Foundation (CHREF) to support the process of transferring a patient from one hospital to another [5].

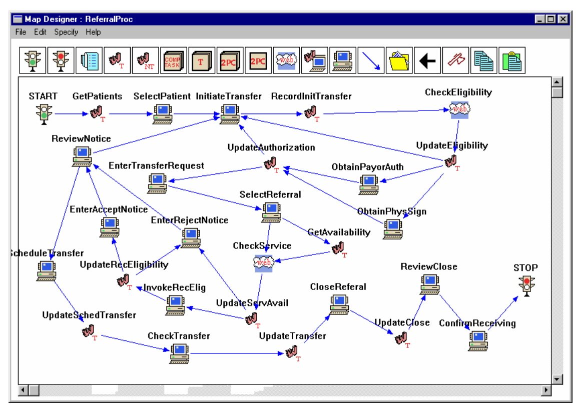

Figure 2. Eligibility referral process

The head ultrasound pathway (Figure 3) is a process that manages initial ultrasound performed when babies arrive at the Neonatal Intensive Care Unit (NICU). The application issues reminders for scheduling tests, retrieving test results, and updating patient records, to the nurse responsible for tracking this data [5].

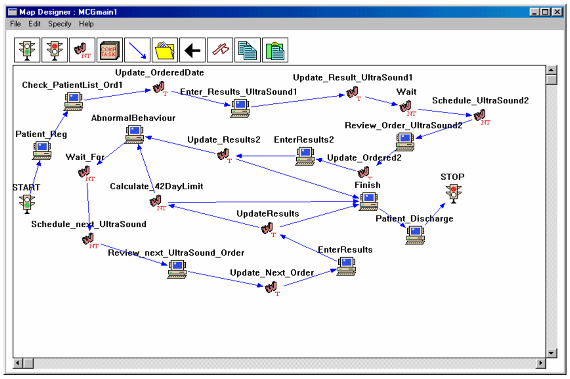

Figure 3. Head ultrasound pathway process

Important questions that can be made relative to the processes illustrated are: “what is the complexity of the processes?”, “can the eligibility referral process be designed in a simpler way?”, and “what areas or regions of the process are more complex and therefore more prone to errors?”

Other important questions are to inquire about the complexity of the two processes. Is the process eligibility referral process more complex that the head ultrasound pathway? Are they both equally complex? Or is the second process more complex that the first one. Complexity measurement gives a method to answer to these questions.

3. Conclusions

The complexity of collaborative and teamwork processes is intuitively connected to effects such as readability, effort, testability, reliability and maintainability of processes. The complexity of a process is also strongly associated with the degree of difficulty a user has to understand and use a process. Therefore, it is important to develop measures to automatically identify complex collaborative and teamwork processes. Afterwards, these processes can be reengineered to reduce the complexity of related activities.

In this paper we discuss fundamental issues related to process complexity. Our objective is to provide the first steps for the development of models and tools for the computation of processes’ complexity.

4. References

[1] A. P. Sheth, W. v. d. Aalst, and I. B. Arpinar, "Processes Driving the Networked Economy," *IEEE Concurrency*, vol. 7, pp. 18-31, 1999.

[2] J. Cardoso, J. Miller, A. Sheth, J. Arnold, and K. Kochut, "Quality of service for workflows and web service processes," *Web Semantics: Science, Services and Agents on the World Wide Web Journal*, vol. 1, pp. 281-308, 2004.

[3] H. Zuse, *A Framework of Software Measurement*. Berlin: Walter de Gruyter Inc, 1997.

[4] IEEE, *IEEE 610, Standard Glossary of Software Engineering Terminology*. New York: Institute of Electrical and Electronic Engineers, 1992.

[5] K. Anyanwu, A. Sheth, J. Cardoso, J. A. Miller, and K. J. Kochut, "Healthcare Enterprise Process Development and Integration," *Journal of Research and Practice in Information Technology, Special Issue in Health Knowledge Management*, vol. 35, pp. 83-98, 2003.