

Approaches to Compute Workflow Complexity

Jorge Cardoso

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

Abstract — During the last 20 years, complexity has been an interesting topic that has been investigated in many fields of science, such as biology, neurology, software engineering, chemistry, psychology, and economy. A survey of the various approaches to understand complexity has lead sometimes to a measurable quantity with a rigorous but narrow definition and other times as merely an ad hoc label. In this paper we investigate the complexity concept to avoid a vague use of the term ‘complexity’ in workflow designs. We present several complexity metrics that have been used for a number of years in adjacent fields of science and explain how they can be adapted and use to evaluate the complexity of workflows.

1 Introduction

The term complexity is commonly found in use throughout many fields of science including biology (Weng, Bhalla et al. 1999), neurology (Koch and Laurent 1999), computer science (Zuse 1991), chemistry (Whitesides and Ismagilov 1999) and economy (Arthur 1999). In the discipline of business process engineering, the terms “complex” or “complexity” are getting some attention (Lakshmanan, Jayaprakash et al. 1991; Cardoso 2005; Cardoso 2005; Cardoso 2005; Cardoso 2005; Rolón, Ruíz et al. 2005; Cardoso 2006; Gruhn and Laue 2006; Gruhn and Laue 2006) and we expect that it will become soon a major research topic.

To study complex workflow and their complexity, one has first answer to the question “What is a complex system?” Several definitions and explanations as to the formal definition of complexity exist, but they all have some aspects in common. According to Lee (Lee 2003), in a naïve sense, complexity is the difficulty in dealing with the system under consideration. Complex systems involve a number of elements, arranged in structure(s), whose dynamics are dominated by nonlinear interactions between their parts. The vast majority of interesting properties of complex systems have to do with nonlinearity and interdependence. Complex systems go through changes and evolution that are not describable by a single rule and include the study of the interactions of the many parts of

the system. Since the interactions of parts are most of time nonlinear, some studies have concentrated their attention in the approximation of complex nonlinear systems by simpler linear ones.

Several researchers have center their efforts on characterizing and quantifying the difficulty associated with complexity (Lee 2003). Indeed, in some cases complexity is defined as the degree of disorder, while in other cases, it is the minimum length of the description of a system or the amount of resource (i.e., time or memory) needed to a system to solve a certain problem. The term complexity is typically used in to characterize real-world entities or systems. For example, we commonly state that "a car is a complex system", but that "a bicycle is less complex than a car". Also, usually, "a large system has large complexity" but "a system with modular design has low complexity."

In the context of Business Process management (BPM) and Workflow Management Systems (WFM), applications and tools have moved from stand-alone office automation systems which augmented individuals, to networked and shared applications. The architectures based on workflow to support this new way of distributed working gave place to a rising complexity. Complexity hampers the correct development of workflows. Some complexity is inherent in the application domain. Some business process are more complex than other due to their domain, while additional complexity is introduced by the strategies and design options used to implement workflows. To manage and control the complexity of workflows and kept it within an acceptable range, it is necessary to develop methods, algorithms and tools to measure their complexity. Measurement is a key to strategic planning. As Lord Kelvin, a noted mathematician once said: "If you can measure something and put a number to it, you can begin to understand it. If you cannot measure it, you have a very sorry ability to understand it..." Indeed, as Burbeck (Burbeck 1996) points out, once complexity has gotten out of control, it takes control. From all the different algorithms and methods that can be devised to measure workflow complexity, each one has its own peculiar features, its strengths and weaknesses. Therefore, in this paper we investigate complexities that arise in the development of workflows and suggest the adaptation of complexity measures from other fields of science to measure the complexity of workflows.

2 Are workflows complex systems?

To answer to the question "Are workflows complex systems?" we need to define and understand what a complex system is. According to several authors, complex systems are systems with the following characteristics:

- Structure – A complex system is a highly structured system, which shows structure with variations (Goldenfeld and Kadanoff 1999)
- Arrangements – Complex systems have a large number of possible arrangements of its parts (Kauffman 1993; Holland 1995; Bar-Yam 1997)
- Interaction – A complex system is one in which there are multiple interactions between many different parts (Rind 1999)
- Inference – Structure and behavior cannot be inferred from the structure and behavior of its parts (Kauffman 1993; Holland 1995; Bar-Yam 1997).

- Response – Parts can adjust in response to changes in adjacent parts (Kauffman 1993; Holland 1995; Bar-Yam 1997).
- Understandability – A complex system is one that by design or function, or both, is difficult to understand and verify (Weng, Bhalla et al. 1999)
- Complexity – Complexity can be a measure of a complex system for characterization and comparison.

In biology, and other related sciences, complex systems have a slightly different sense. The term complex system refers to a system with similar characteristics with the one we have given but the parts are coupled in a nonlinear fashion and are described using mathematical models from dynamic systems, which include differential equations.

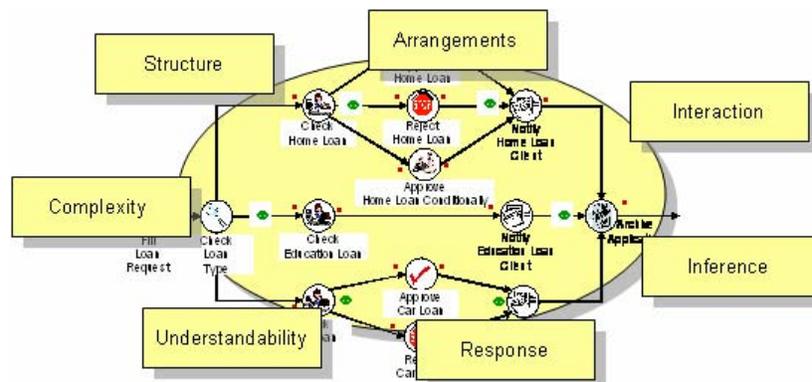


Figure 1. A workflow as a complex system

These definitions for the term complex system allow us to conclude that workflows are indeed complex systems. Let us see how this conclusion can be reached. By definition, workflows are an ordered (i.e., structure) group of business activities (i.e., components or part of a system) undertaken by an organization to achieve a goal. Activities take one or more kind of inputs and create outputs. The outputs of some activities are the inputs or other activities (i.e., interaction). A workflow rarely operates in isolation, i.e. other workflows will depend on it and it will depend on other workflows (i.e., interaction). The measures presented in (Lakshmanan, Jayaprakash et al. 1991; Cardoso 2005; Cardoso 2005; Cardoso 2005; Cardoso 2005; Rolón, Ruíz et al. 2005; Cardoso 2006; Gruhn and Laue 2006; Gruhn and Laue 2006) can be used to characterize and compare workflows (i.e., complexity). Some workflows have a high number of activities and an elaborated design (i.e., understandability). Examples include workflows that model receiving orders, marketing services, selling products, delivering services, distributing products, and invoicing for services. Additionally, knowing the structure and behavior of each activity does not allow deducing the structure and behavior of a workflow (i.e., inference). Adaptation technique (Hermann 1995; Miller, Cardoso et al. 2002; Adams, Edmond et al. 2003; Cardoso and Sheth 2005) can be used to change the structure of a workflow and its activities when it does not meet initial requirements (i.e., response). A workflow links a set of activities with a specific start and finish point that describe a process, but there is most of the time more than one way to arrange the activities of a workflow to achieve an organizational goal (i.e., arrangements).

3 Workflow complexity

According to the Merriam-Webster dictionary, complexity is: a) something complex and b) the quality or state of being complex. These definitions take us back to the concept of complex. The same dictionary defines the term complex as something composed of two or more parts or which is hard to separate, analyze, or solve. These particular definitions are not too useful, since they are rather vague to define precisely what workflow complexity is.

Bar-Yam (Bar-Yam 1997) states that the complexity of a system is strongly connected to its number of possible states and the information needed to describe the system. This definition is attractive, but when applied to workflows it is too restrictive. We believe that several complexity metrics can be devised based on the structure, parts and organization of workflows. This means that two distinct workflows, possibly with the same number of states and that require the same amount of information to describe them, can have a different complexity.

Christopher Alexander (Alexander 1979) associates complexity with the relations among the entities which interact and the result of those relations. Alexander's notion of complexity considers the notion of "order." Order takes use to the idea of dynamic system. It is associated with the action of doing something. On the other hand, we can also consider that complexity can be passive, where the order of states is not taken into account but only the properties of states. As a result, the order of a system, which describes the relationships among the parts of a system, can be a complexity a measure, but the individual parts of the systems can also contribute to complexity and, therefore, need to be measured.

We believe that from the different definitions that have been given for the term complexity, the most adequate ones come from the field of software engineering. Several definitions have been given to describe the meaning of software complexity (Zuse 1997). After analyzing the characteristics and specific aspects of workflows, we believe that the definition that is better suited to describe workflow complexity can be derived from the Standard Glossary of Software Engineering Terminology (IEEE 1992). Therefore, we define workflow 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 workflow characteristics.

4 Ordered, structured and random workflows

When we view workflows as complex systems, we are interested in their complexity. We can identify three types of workflows that have a distinct complexity based on their structure: ordered, structured, and random workflows. Ordered workflows are perceived as simple, while random workflows are at the other end of the complexity scale, whatever that scale may be. Structured workflows are deemed to lie somewhere in between. From Figure 2 we can see that complexity can be related to the general idea of a relationship between structure and complexity, as the consensus is that random workflows do not possess any structure.

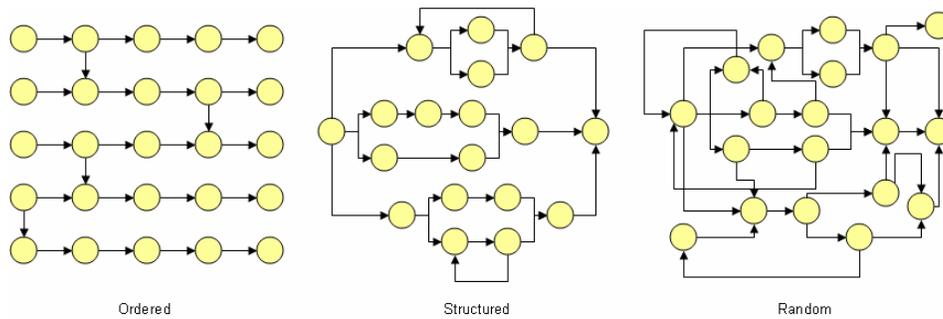


Figure 2. Ordered, structured and random workflows

Workflows are not static complex systems. They are constantly undergoing revisions, adaptations, changes, and modifications to meet end users needs. The complexity of workflows tends to increase as they are maintained over a period of time. Workflows may even require to be reengineered when their behavior is no longer satisfactory or their complexity has reached an upper threshold. Indeed, a workflow that is initially ordered can easily, with time, change and become a structured one, if no precautions are taken, it can even become a random workflow. The typical evolution of workflows is illustrated in Figure 3.

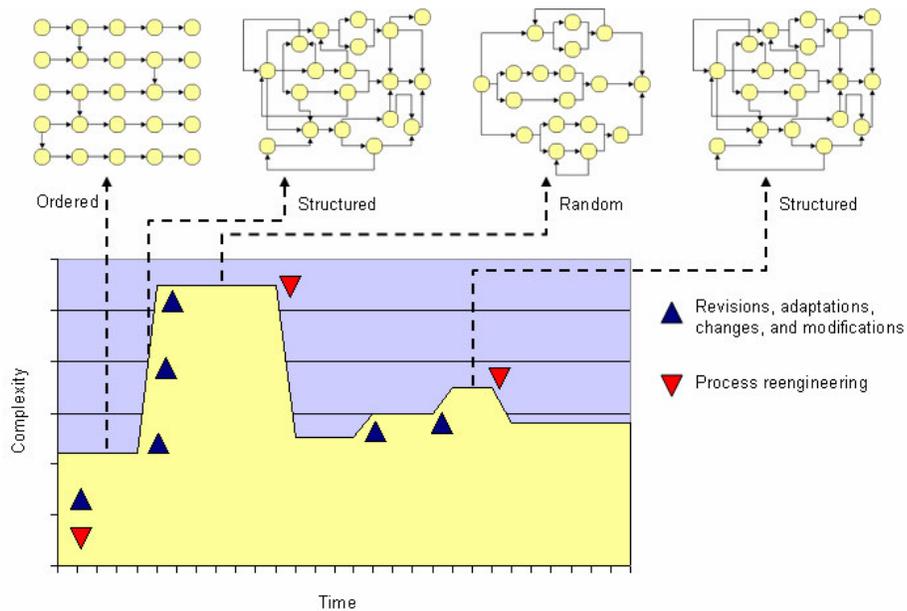


Figure 3. Workflow evolution with time.

5 Perspectives on Workflow Complexity

Workflow specifications can be understood from a number of different perspectives. Aalst and Hofstede (Aalst and Hofstede 2005) discuss three main perspectives: control-flow, data, and resource perspectives. We extend this categorization adding the activity perspective. This perspective views activities in isolation without taking into account the

control-flow that interconnects activities. It will become clear that this perspective is very useful to evaluate the complexity of workflows. For each perspective, one or more complexity metric can be derived. Therefore, four main complexity perspectives can be identified: activity complexity, control-flow complexity, data-flow complexity, and resource complexity. The reader should realize that these perspectives are not complete.

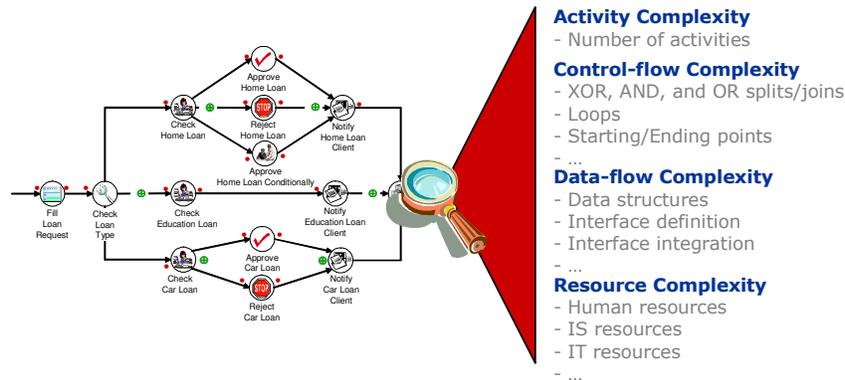


Figure 4. Perspective top workflow complexity

Activity complexity. This view on complexity simply calculates the number of activities a process has. While this complexity metric is very simple, it is very important to complement other forms of complexity. The control-flow complexity of a process can be very low while its activity complexity can be very high. For example, a sequential process that has a thousand activities has a control-flow complexity of 0, whereas its activity complexity is 100. This metric was inspired by lines-of-code (LOC) metric used with a significant success rate in software engineering (Jones 1986).

Control-flow complexity. The control-flow perspective describes activities and their execution ordering through different constructors, which permit flow of execution control. Constructors include sequence, choice, parallelism, splits, joins, loops, and ending and starting points (Cardoso 2005). Splits allow defining the possible control paths that exist in a 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 needs to take into account the existence of XOR-split/join, OR-split/join, AND-split/join, loops, etc.

Data-flow complexity. This perspective is layered on top of the control perspective. Documents and other data objects flow between activities. Local and global variables of the workflow are used in effect pre- and post-conditions of activity execution (Aalst and Hofstede 2005). The data-flow complexity of a process increases with the complexity of its data structures, the number of formal parameters of activities, and the mappings between activities' data (Reijers and Vanderfeesten 2004). A data-flow complexity metric can be composed of several sub-metrics which include: data complexity, interface complexity, and interface integration complexity (Cardoso 2005). While the first two sub-metrics are related

to static data aspects (data declaration), the third metric is more dynamic in nature and focuses on data dependencies between the different activities of a process.

Resource complexity. The resource perspective provides an organizational structure anchor to the workflow in the form of human and device roles responsible for executing activities (Aalst and Hofstede 2005). Activities in a process need to access resources during their executions. A resource is defined to be any entity (e.g. human resources, IS resources, and IT resources) required by an activity for its execution, such as an external document, a database, a printer, an external application, or role (Du, Davis et al. 1999; zur Mühlen 1999). Resources, such as actors and roles, can be structured in the context of an organization. The structure that is used to shape the different types of resources can be analyzed to determine its complexity. This analysis can help managers to lower administrative costs and better optimize resource utilization.

6 Complexity metrics for workflows

The overall goal of workflow complexity analysis is to improve the comprehensibility of workflows. The graphical representation of most workflow specification languages provides the user with the capability to recognize complex areas of workflows. Thus, it is important to develop methods and measurements to automatically identify complex workflows and complex areas of workflows. Afterwards, these workflows can be reengineered to reduce their complexity. Because of the difficulty involved in determining a single universal rating number to describe the complexity of workflows, it is necessary to develop multi-dimensional metrics of complexity and then validate the metrics through objective and subjective measurements.

In this section we provide a theoretical survey of complexity considerations and metrics from several fields and we relate them to workflow modeling. We view the complexity of a workflow in different dimensions, namely information complexity, cyclomatic complexity, Kolmogorov complexity, cognitive complexity, and computational complexity. A further empirical investigation might ultimately lead to establishing a complexity theory for workflow models.

6.1 Information theory

Information theory is a statistical theory concerning the storage and transmission of data by dealing with the limits and efficiency of information processing. Claude Shannon was the first in 1948 to propose the use of binary digits for coding information. Information theory has led to the introduction of the concept of entropy (also called Shannon's entropy). The concept of entropy describes how much randomness, disorder or uncertainty there is in a closed system. An alternative way to look at this is to talk about how much information is present in a system. Entropy is defined in terms of a discrete random event x , with possible states $1..n$ as:

$$H(x) = -\sum_{i=1}^n p(i) \log_2 p(i)$$

The concept of entropy can be applied to workflows since we consider them to be complex systems.

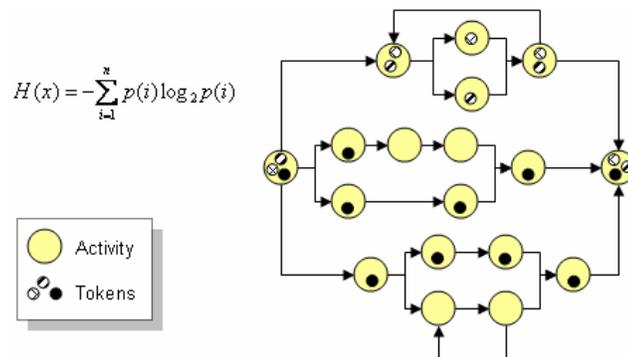


Figure 5. Entropy and workflow complexity

Let us consider the workflow in Figure 5. A token present in an activity indicates that the activity has been invoked during the enactment of the workflow. Tokens with the same pattern belong to the same instance. Obviously some activities have more tokens than others. We can compute the runtime complexity using the entropy function by considering $p(i)$ to be the probability of activity i to have a token. The frequency of tokens in some activities is not very high (e.g. activities with only one token), while other have a high frequency (e.g. the initial and final activity). Calculating the entropy can obtain a measure of the disorder or randomness of workflow instances. It should be noticed that is a runtime complexity metric.

6.2 Kolmogorov complexity

Kolmogorov complexity can also be used to compute the complexity of a workflow. By definition, the Kolmogorov complexity of an object is the length of the shortest binary program run on a universal Turing Machine (TM) that can reproduce the object. Formally, assuming that we have a string description of a Turing machine M , denoted $\langle M \rangle$, an input i and a string s , then the concatenated string $\langle M \rangle i$ is a description of s . Several descriptions for s exist. Among all the descriptions there is one with shortest length denoted $d(s)$. The minimal length of the minimal description of s is expressed with function $K(s)$, i.e.,

$$K(s) = |d(s)|$$

As a result, the amount of information in a string is the size of the shortest program that outputs the string. Kolmogorov complexity has been connected with Information Theory and proved to be closely related to Claude Shannon's entropy rate of an information source.

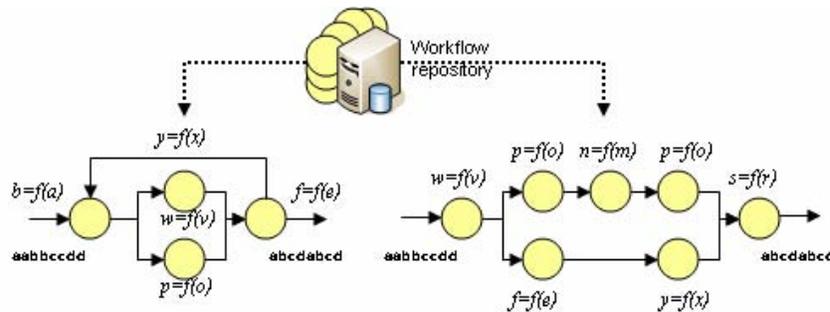


Figure 6. Kolmogorov complexity of workflows

Let us assume that we have a workflow repository with n activities (a_i). Each activity is represented with function $y_i = f_i(x_i)$ as shown in Figure 6. The challenge that we propose is to compose a workflow – with the help of the activities available in the repository – which outputs the string *abcdabcd* when the string *aabbccdd* is given as an input. Additionally, we are looking for the workflow that has a minimal length description according to Kolmogorov complexity. In Figure 6 two such workflows are illustrated. As it can be seen the workflow from the left has a lower length description. One of the important characteristics of the theory of Kolmogorov complexity stems from this independence of the description method. For workflows this means that the complexity is independent of the workflow used to model a process.

6.3 Cyclomatic complexity

In our work previous work (Cardoso 2005; Cardoso 2005) we have designed a processes complexity metric that borrowed some ideas from McCabe’s cyclomatic complexity. Our objective was to develop a metric that could be used in the same way as the MCC metric but to evaluate processes' complexity.

One of the first important observations that can be made from the MCC control flow graph, shown in Figure 7, is that this graph is extremely similar to a process. One major difference is that the nodes of a MCC control flow graph have identical semantics, while process nodes (i.e., activities) can have different semantics (e.g., AND-splits, XOR-splits, OR-joins, etc). Our approach has tackled this major difference.

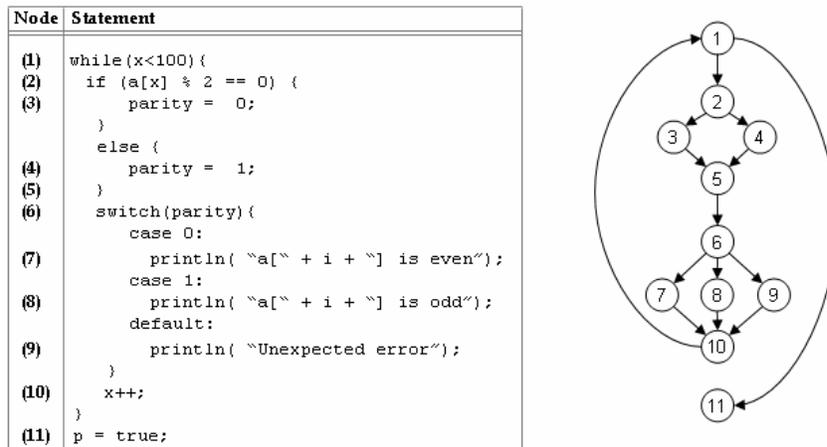


Figure 7. Cyclomatic complexity

The metric that we have previously developed and tested, called Control-flow Complexity (CFC) metric, was based on the analysis of XOR-splits, OR-splits, and AND-splits control-flow elements. The main idea behind the metric was to evaluate the number of mental states that have to be considered when a designer is developing a process. Splits introduce the notion of mental states in processes. When a split (XOR, OR, or AND) is introduced in a process, the business process designer has to mentally create a map or structure that accounts for the number of states that can be reached from the split. The notion of mental state is important since there are theories (Miller 1956) suggesting that complexity beyond a certain point defeats the human mind's ability to perform accurate symbolic manipulations, and hence results in error.

Mathematically, the control-flow complexity metric is additive, thus it is very easy to calculate the complexity of a process, by simply adding the CFC of all split constructs. The control-flow complexity was calculated as follows, where P is a process and a an activity.

$$\begin{aligned}
 CFC(P) = & \sum_{a \in P, a \text{ is a xor-split}} CFC_{XOR}(a) \\
 & + \sum_{a \in P, a \text{ is a or-split}} CFC_{OR}(a) + \sum_{a \in P, a \text{ is a and-split}} CFC_{AND}(a)
 \end{aligned}$$

The higher the value of $CFC_{XOR}(a)$, $CFC_{OR}(a)$, and $CFC_{AND}(a)$, the more complex is a process design, since developer has to handle all the states between control-flow constructs (splits) and their associated outgoing transitions and activities. Each formula to calculate the complexity of a split construct is based on the number of states that follow the construct. CFC analysis seeks to evaluate complexity without direct execution of processes.

The advantages of the CFC metric is that it can be used as a maintenance and quality metric, it gives the relative complexity of process designs, and it is easy to apply. Disadvantages of the CFC metric include the inability to measure data complexity, only control-flow complexity is measured. Additionally, the same weight is placed on nested and non-nested loops. However, deeply nested conditional structures are harder to understand than non-nested structures.

6.4 Cognitive complexity

Cognitive complexity is related to cognitive psychology that aims at studying, among other things, thinking, reasoning, and decision making. Cognitive complexity focuses on the analysis of how complicated a problem is from the perspective of the person trying to solve it. For workflow designers, the ability of coping with complexity is a fundamental issue and influences the quality of the final product.

Cognition involves both short-term and long-term memories. Most definitions of short-term memory limit the duration of storage to less than about 30 seconds. On the other hand, long-term memory can last as little as 30 seconds or as long as decades. Understanding how short-term and long-term memories are used can lead to workflow metrics that predict human performance and comprehension in workflow design, and can be used to assess and improve the understandability of workflows.

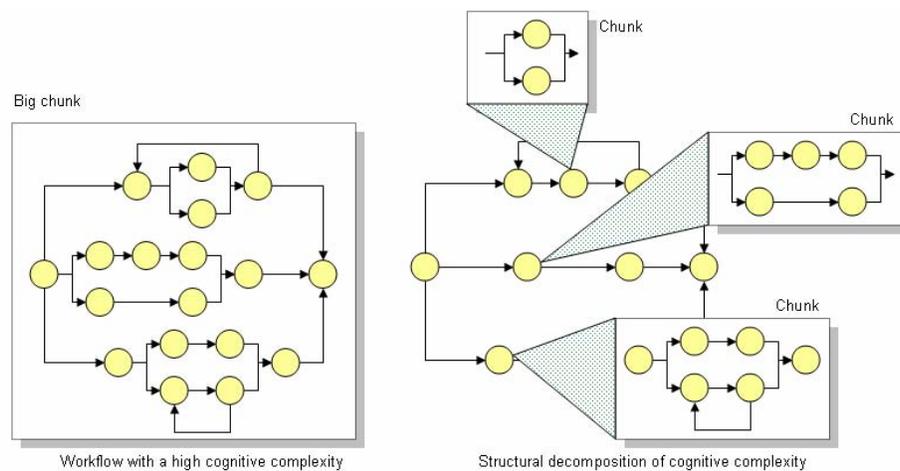


Figure 8. Decreasing the cognitive complexity of workflows

For example, let us consider the workflow on the left side of Figure 8. The workflow is one structure of 18 activities. Each activity has a name and a perceived behavior. When a workflow designer looks at the workflow its brain tries to group all the concepts associated with the activities together into one big abstract unit called chunk (Cant, Jeffery et al. 1995). Unfortunately, the short-term memory is a container where only a small finite number of concepts can be stored. Miller (Miller 1956) argued that human short-term memory has a forward memory span of approximately seven items plus or minus two. This means that the number of items, for example, words or numbers, that a person can retain and recall is 7 ± 2 . In our example, too many concepts are associated with the workflow and, therefore, they cannot all be stored in the short-term memory container. As a result, when confronted with workflow design decisions, designers have a higher risk of making errors. Indeed, when short-term memory is overloaded, errors occur (Kintsch 1998).

One solution to enable workflow designers to avoid making errors while working with the workflow on the left side of Figure 8 is to hierarchically decompose the workflow, i.e., the workflow is broken into a hierarchy of sub-workflows. Using this strategy, chunks become composed of chunks and chunks can become part of long-term memory if they are

encountered often enough. Once in long-term memory, they can be recognized when they are encountered. In computer programs, it has been found that the identifier density is a good predictor of error provided time is constrained (Klemola 1999). In our context, activities can be thought as identifiers.

Furthermore, Athey (1976) and Ören (1978) have noticed that for a problem with a low complexity, an individual may need a low level of information processing to cope with the decisions to solve the problem. If the complexity of the problem increases, its information processing level may increase. However, for each individual there is a critical point beyond which the level of processed information effectiveness is decreased. After the critical point, an increase in the complexity of a problem may worsen the individual's ability to cope with complexity, by causing a decrease level of information processing. As a result, workflows should have a structure within predefined levels of complexity. The workflow on the left side of Figure 8 may have a complexity that requires a level of information processing that is beyond critical point making the individual's ability to cope with complexity decreases.

6.5 Computational complexity theory

Computational complexity theory studies the resources of the computation required to solve a given problem. Resources usually involve parameters such as time and space. Time complexity studies the length of time it takes to find a solution or complete a process. It is a function of the size of the input. Space complexity studies the amount of physical storage required for a system to perform a certain operation. It is also a function of the size of the input.

The basic property of workflow is that they are case-based (Aalst 1998). This means that every activity is executed for a specific case. The concept of time complexity theory can be used to study the number of activities that are needed to complete a particular workflow instance. In the same way, space complexity can also be used to study the storage space needed to complete and instance.

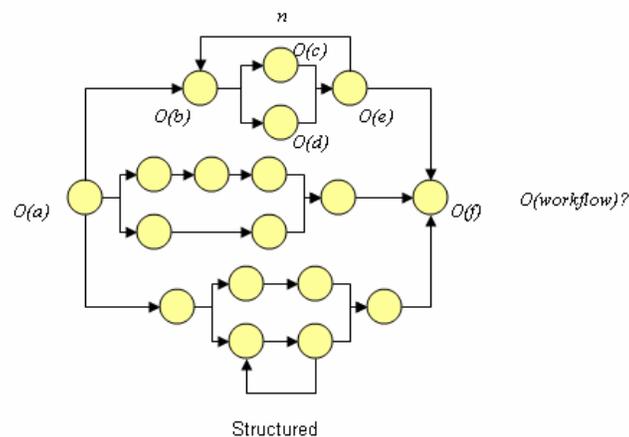


Figure 9. Time complexity of a workflow

Let us consider the workflow in Figure 9. Each activity has a Big O notation that describes its asymptotic behavior. Some transitions are loops that are taken n times. Having these settings, what is the time computational complexity of the workflow? The difficulty

of computing the overall complexity is due to the existence of parallel constructs and especially non-deterministic splits.

7 Conclusions

Several workflow specification languages and systems have been developed to ease the task of modeling and supporting business processes. Emergent business-to-business and business-to-customer distributed systems will most likely be based on workflow management platforms. One step in this direction was the specification of BPEL, a language to design processes that orchestrate Web services. As organization adopt this new technological way of doing business, they will rapidly wish to have available tools to guarantee that the developed workflows are simple, modular, easy to understand, easy to maintain and easy to re-engineer. To achieve these objectives, one can calculate the complexity of workflows. The complexity of workflows is intuitively connected to effects such as readability, understandability, effort, testability, reliability and maintainability. As known from software engineering, it is a fact that it is cost-effective to fix a defect earlier in the design lifecycle than later. While these characteristics are fundamental in the context of workflow, and business processes in general, only a few contributions have targeted these problems.

In this paper, we surveyed several complexity approaches developed in other fields of science and we discussed to what extent these metrics and measurements can be adapted for workflows. While more complexity approaches exist, we have limited our scope to the one that we feel can have a greater impact on workflow management. Namely, we have described and illustrated possible scenarios that explain how information complexity, cyclomatic complexity, Kolmogorov complexity, cognitive complexity, and computational complexity can be used to evaluate the complexity of workflows.

A further empirical investigation of the application of these approaches to evaluate complexity might ultimately lead to establishing a complexity theory of business process models. In order to demonstrate that these metrics serves their purpose, we plan to carry out several empirical validations by means of controlled experiments and develop tools that will be made available to workflow designers.

8 References

- Aalst, W. M. P. v. d. (1998). "The Application of Petri Nets to Workflow Management." The Journal of Circuits, Systems and Computers **8**(1): 21-66.
- Aalst, W. M. P. v. d. and A. H. M. t. Hofstede (2005). "YAWL: Yet Another Workflow Language." Information Systems Frontiers **30**(4): 245-275.
- Adams, M., D. Edmond, et al. (2003). The application of Activity Theory to dynamic workflow adaptation issues. 7th Pacific Asia Conference on Information Systems (PACIS 2003), Adelaide, South Australia.
- Alexander, C. (1979). The Timeless Way of Building, Oxford University Press.
- Arthur, W. B. (1999). "Complexity and the Economy." Science **284**(5411): 107-109.

- Athey, T. H. (1976). Training the Systems Analysts to Solve Complex Real World Problems. 14th Annual Computer Personnel Research Conference, ACM Press.
- Bar-Yam, Y. (1997). Dynamics of Complex Systems, Perseus.
- Burbeck, S. (1996). "Real-time complexity metrics for smalltalk methods." IBM Systems Journal: 1-28.
- Cant, S., D. R. Jeffery, et al. (1995). "A conceptual model of cognitive complexity of elements of the programming process." Inf. Software Technol. **37**(7): 351-362.
- Cardoso, J. (2005). About the Complexity of Teamwork and Collaboration Processes. IEEE International Symposium on Applications and the Internet (SAINT 2005), Workshop - Teamware: supporting scalable virtual teams in multi-organizational settings, Trento, Italy, IEEE Computer Society.
- Cardoso, J. (2005). About the Data-Flow Complexity of Web Processes. 6th International Workshop on Business Process Modeling, Development, and Support: Business Processes and Support Systems: Design for Flexibility, Porto, Portugal.
- Cardoso, J. (2005). Control-flow Complexity Measurement of Processes and Weyuker's Properties. 6th International Conference on Enformatika, 26-28, October 2005, Budapest, Hungary, International Academy of Sciences.
- Cardoso, J. (2005). Evaluating Workflows and Web Process Complexity. Workflow Handbook 2005. L. Fischer. Lighthouse Point, FL, USA, Future Strategies Inc.: 284-290.
- Cardoso, J. (2006). "Complexity Analysis of BPEL Web Processes." Journal of Software Process: Improvement and Practice, InterScience, Wiley (**to appear**).
- Cardoso, J. and A. Sheth (2005). Adaptation and Workflow Management Systems. International Conference WWW/Internet 2005, Lisbon, Portugal.
- Du, W., J. Davis, et al. (1999). Enterprise workflow resource management. International Workshop on Research Issues in Data Engineering, Sydney, Australia.
- Goldenfeld, N. and L. P. Kadanoff (1999). "Simple Lessons from Complexity." Science **284**(5411): 87-89.
- Gruhn, V. and R. Laue (2006). Adopting the Cognitive Complexity Measure for Business Process Models. 5th IEEE International Conference on Cognitive Informatics, Beijing, China.
- Gruhn, V. and R. Laue (2006). Complexity Metrics for Business Process Models. 9th International Conference on Business Information Systems, Klagenfurt, Austria.
- Hermann, T. (1995). Workflow Management Systems: Ensuring Organizational Flexibility by Possibilities of Adaptation and Negotiation. Proceedings of the Conference on Organizational Computing Systems (COOCS'95).
- Holland, J. H. (1995). Hidden Order: How Adaptation Builds Complexity, Reading: Addison-Wesley.
- IEEE (1992). IEEE 610, Standard Glossary of Software Engineering Terminology. New York, Institute of Electrical and Electronic Engineers.
- Jones, T. C. (1986). Programming Productivity. New York, McGraw-Hill.

- Kauffman, S. A. (1993). The Origins of Order: Self-Organization and Selection in Evolution. New York, Oxford University Press.
- Kintsch, W. (1998). Comprehension: a paradigm for cognition, Cambridge University Press.
- Klemola, T. (1999). Software Comprehension: Theory and Metrics, Concordia University.
- Koch, C. and G. Laurent (1999). "Complexity and the Nervous System." Science **284**(5411): 96-98.
- Lakshmanan, K. B., S. Jayaprakash, et al. (1991). "Properties of Control-Flow Complexity Measures." IEEE Transactions on Software Engineering archive **17**(12): 1289 - 1295.
- Lee, T. (2003). Complexity Theory in Axiomatic Design. Department of Mechanical Engineering. Massachusetts, Massachusetts Institute of Technology: 182.
- Miller, G. A. (1956). "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information." Psychological Review **63**: 81-97.
- Miller, J. A., J. S. Cardoso, et al. (2002). Using Simulation to Facilitate Effective Workflow Adaptation. Proceedings of the 35th Annual Simulation Symposium (ANSS'02), San Diego, California.
- Ören, T. I. (1978). Rationale for Large Scale Systems Simulation Software based on Cybernetics and General Systems Theories. Cybernetics and Modelling and Simulation Large Scale Systems, Namur, Belgium.
- Reijers, H. A. and I. T. P. Vanderfeesten (2004). Cohesion and Coupling Metrics for Workflow Process Design. BPM 2004 (LNCS 3080). J. Desel, B. Pernici and M. Weske. Berlin, Heidelberg, Springer-Verlag. **LNCS 3080**: 290-305.
- Rind, D. (1999). "Complexity and Climate." Science **284**(5411): 103-107.
- Rolón, E., F. Ruíz, et al. (2005). "Aplicación de métricas software en la evaluación de modelos de procesos de negocio." Revista Sociedad Chilena de Ciencia de la Computación **6**(1).
- Weng, G., U. S. Bhalla, et al. (1999). "Complexity in Biological Signaling Systems." Science **284**(5411): 92-96.
- Whitesides, G. M. and R. F. Ismagilov (1999). "Complexity in Chemistry." Science **284**(5411): 89-92.
- zur Mühlen, M. (1999). Resource Modeling in Workflow Applications. Workflow Management Conference, Muenster, Germany.
- Zuse, H. (1991). Software Complexity: Measures and Methods. Berlin, New York, DeGruyter Publisher.
- Zuse, H. (1997). A Framework of Software Measurement. Berlin, Walter de Gruyter Inc.