Theoretical vs. Practical Complexity: The Case of UML

Keng Siau, University of Nebraska-Lincoln, USA
John Erickson, University of Nebraska at Omaha, USA
LihYunn Lee, University of Nebraska-Lincoln, USA

ABSTRACT

Unified Modeling Language (UML) is the standard visual modeling language for Object Oriented (OO) systems development, but it has been criticized for its complexity, inconsistent semantics, and ambiguous constructs. A set of complexity indices for UML and the nine diagramming techniques in UML was compiled recently. The complexity analysis is formulated based on the number of constructs, associations, roles, and so forth, in a modeling method. We argue that this set of metrics provides an indication of the theoretical complexity of the modeling methods. On the other hand, the theoretical complexity of the modeling methods does not necessarily relate to the practical complexity. We hypothesize that UML’s complexity is not as daunting as the metrics imply, because not all the constructs are used all the time. Thus, in addition to theoretical complexity, a set of metrics for estimating practical complexity can be developed, based on the most commonly used constructs (instead of all constructs). In this research, we use secondary data to test our hypothesis that practical complexity is different from theoretical complexity.

Keywords: cognition; complexity; metrics; unified modeling language

INTRODUCTION

Over the past 10 to 15 years, object-oriented systems analysis and design methodology have emerged from object-oriented programming. Over time, separate modeling methods merged to form a combined analysis and design technique called the Unified Modeling Language (UML) (Booch et al., 1999). UML consists of nine distinct diagramming techniques to support object oriented systems development. Despite the standardization of UML by the Object Management Group, researchers and practitioners criticize UML’s complexity and the ambiguity of its constructs (Siau & Lee, 2005; Siau & Loo, 2002; Siau et al., 2002). Siau and Cao (2001) used the complexity metrics developed by Rossi and Brinkkemper (1996) to analyze the diagram-
ming techniques in UML and compared them to other modeling methods.

In this research, we argue that the complexity metrics developed by Rossi and Brinkkemper (1996), and adopted by Siau and Cao (2001) (1) present the theoretical complexity of the modeling techniques and (2) present the theoretical upper limit of the complexity, because the metrics were formulated based on the total number of objects, relationships, and property types in the modeling techniques. We define theoretical complexity as the maximum upper limit of the complexity of a modeling technique, taking into account all its objects, relationships, and property types. Practical complexity, on the other hand, takes into consideration that not all modeling constructs will be used simultaneously, and these various constructs have different importance, and, therefore, they should not be weighted equally. Based on these premises, practical complexity must be less than or equal to theoretical complexity.

We aim to extend the complexity metrics to take into account the actual usage/practice of the modeling method. In other words, in addition to the theoretical complexity, we should be able to provide an estimation of the practical complexity of a modeling method, based on the typical usage of the modeling method. The practical complexity of a modeling language will provide a more realistic estimation and will complement the theoretical complexity. UML has been criticized for being overly complex, and the theoretical complexity metrics bear this out; however, if the UML that people learn and use does not make use of all constructs all the time, then perhaps it is not out of line to conjecture that UML, while undoubtedly complex, is not as complex and difficult to learn and use as the metrical analyses completed to date indicate. If so, this research can be used to suggest better ways of learning and using UML.

In this paper, we test our hypothesis that there exists a practical complexity that is different from the theoretical complexity. This is operationalized as the time needed to interpret a set of use-case and class diagrams, testing that the time required for the two diagrams is different from the theoretical estimation. Although there are many ways to measure complexity, we propose that difficulties that subjects encounter during diagram interpretation will manifest in longer interpretation times, and, therefore, time is one means of measuring complexity. The validity of using interpretation times as a surrogate of complexity is discussed in the latter part of the paper.

The rest of the paper is organized as follows. The next section presents the literature of the area, focusing on complexity and UML. This is followed by the Theoretical Foundations section, Research Question and Hypothesis section, Research Methodology and Design section, Results and Discussion section, and, finally, the Conclusion and Future Research section.

LITERATURE REVIEW

Complexity can take on many forms. For the purposes of this research, complexity will be approached from two separate but closely related perspectives: cognitive complexity, as related to human perception; and structural complexity, as related to the structural properties of the diagramming techniques found in modeling approaches such as UML diagrams. In this context, cognitive complexity can be defined as the mental burden people face as they work with systems development constructs.

Further, in addition to the theoretical basis of cognitive complexity detailed in the
Distributed Database Design for Mobile Geographical Applications
Manhoi Choy, Mei-Po Kwan and Hong V. Leong (2000). *Journal of Database Management* (pp. 3-15).
www.igi-global.com/article/distributed-database-design-mobile-geographical/3245?camid=4v1a

The MAIS Approach to Web Service Design
www.igi-global.com/chapter/mais-approach-web-service-design/4393?camid=4v1a