Available online at www.sciencedirect.com
SCIENCE DIRECT
Information and Software Technology 47 (2005) 675–692
INFORMATION AND SOFTWARE TECHNOLOGY
www.elsevier.com/locate/infsof

Quality and comprehension of UML interaction diagrams-an experimental comparison

Chanan Glezer*, Mark Last, Efrat Nachmany, Peretz Shoval

Department of Information Systems Engineering, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel

Received 29 September 2004; revised 24 December 2004; accepted 10 January 2005
Available online 11 March 2005

Abstract

UML (Unified Modeling Language) is a collection of somewhat overlapping modeling techniques, thus creating a difficulty in establishing practical guidelines for selecting the most suitable techniques for modeling OO artifacts. This is true mainly with respect to two types of interaction diagrams: *Sequence* and *collaboration*. Attempts have been made to evaluate the comprehensibility of these diagram types for various types of applications, but they did not address the issue of quality of diagrams created by analysts. This article reports the findings from a controlled experiment where both the comprehensibility and quality of the interaction diagrams were investigated in two application domains: management information systems (MIS) and real-time (RT) systems.

Our results indicate that collaboration diagrams are easier to comprehend than sequence diagrams in RT systems, but there is no difference in comprehension of the two diagram types in MIS. Irrespective of the diagram type, it is easier to comprehend interaction diagrams of MIS than of RT systems. With respect to diagram quality, in the case of MIS, analysts create collaboration diagrams of better quality than sequence diagrams, but there is no significant difference in quality of diagrams created in RT systems. Irrespective of the diagram type, more correct diagrams are created in MIS applications than in RT applications.

Keywords: Collaboration diagrams; Experiments; Interaction diagrams; Method/model evaluation; Object-oriented analysis; Sequence diagrams; Software engineering; UML.

1. Introduction

The Unified Modeling Language (UML) accommodates several types of diagrams in an attempt to model the structural and dynamic aspects of systems. The broad acceptance of UML as a language for Object Oriented (OO) analysis and design has fundamentally been based on its usefulness as a means of communication among all parties involved in the application development process [1]. Moreover, the prominent OMG Model Driven Architecture (MDA) initiative [2], which enables interoperability of applications across middleware boundaries, is already adopted by many companies in a plethora of application domains. Nevertheless, the nature of UML, being a collection of somewhat overlapping modeling techniques, and its recent proliferation in heterogeneous application domains, creates a difficulty in establishing practical guidelines for selecting the most suitable UML techniques for modeling OO artifacts.

Methodologies in general (and UML in particular) can be evaluated and compared on various dimensions, i.e. quality of the analysis or design products, comprehensibility, learnability, ease of use, user/developer satisfaction or preference, and more [3]. Criteria such as: degree of chunking and fragmentation of requirements, the number of simultaneous items or semantic processing required for creating a schema, and more are proposed in order to promote the performance and readability of conceptual modeling in terms of performance and readability [4].

Several attempts have been made to evaluate the semantic comprehension of object-oriented diagrams. Diagrammatic reasoning with multiple diagrams is comprised of searching for related information (perceptual reasoning), and developing hypotheses about the target system

* Corresponding author. Tel.: +972 8 6472239; fax: +972 8 6472239.
E-mail address: chanan@bgu.ac.il (C. Glezer).

0950-5849/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.infsof.2005.01.003

(conceptual reasoning) [5]. The findings from a controlled experiment indicate that perceptual and conceptual integration processes are facilitated by incorporating visual cues and contextual information in the multiple diagrams as representation aids [5]. Otero and Dolado [1,6] studied the use of interaction diagrams in UML for applications of varying complexity, ranging from management information systems, through embedded systems, to real-time reactive systems. Their work focused on evaluating comprehensibility of diagrams, assuming that a fit between the application domain and diagram type will positively impact the level of semantic comprehension of a given system's specification and, in turn, will indirectly facilitate the use of the proper diagrams in system construction.

No attempt was made, however, to directly investigate the quality of interaction diagrams produced in various scenarios under an objective yardstick, a lacuna which the current study attempts to address. Thus, this article reports the findings from a controlled experiment conducted in a university setting, where both the semantic comprehension and quality of UML interaction diagrams was investigated for various application domains.

The rest of the paper is organized as follows: Section 2 discusses UML and interaction diagrams. Section 3 describes related work on previous empirical comparisons of software modeling frameworks in general, and specifically of UML interaction diagrams. Section 4 describes our experiment, including the dependent and independent variables, the research hypotheses, the participants, the documents included in the experiment, and the experimental design. It also discusses threats to validity and how we treated them. Section 5 presents the results in terms of diagrams comprehensibility, quality of diagram construction, time taken to complete the comprehension and the construction tasks, perceived comprehensibility and perceived ease of construction. Section 6 discusses the results of the experiments in light of the hypotheses and related work, and finally, Section 7 outlines the limitations of this study and suggests topics for future research.

2. UML and interaction diagrams

2.1. The unified modeling language (UML)

As strategic value of software increases for many companies, the industry is eagerly looking for techniques that automate the production of software in an attempt to improve quality, reduce costs and expedite time-to-market. These techniques include, among others: component technology, visual programming, patterns, and modeling frameworks.

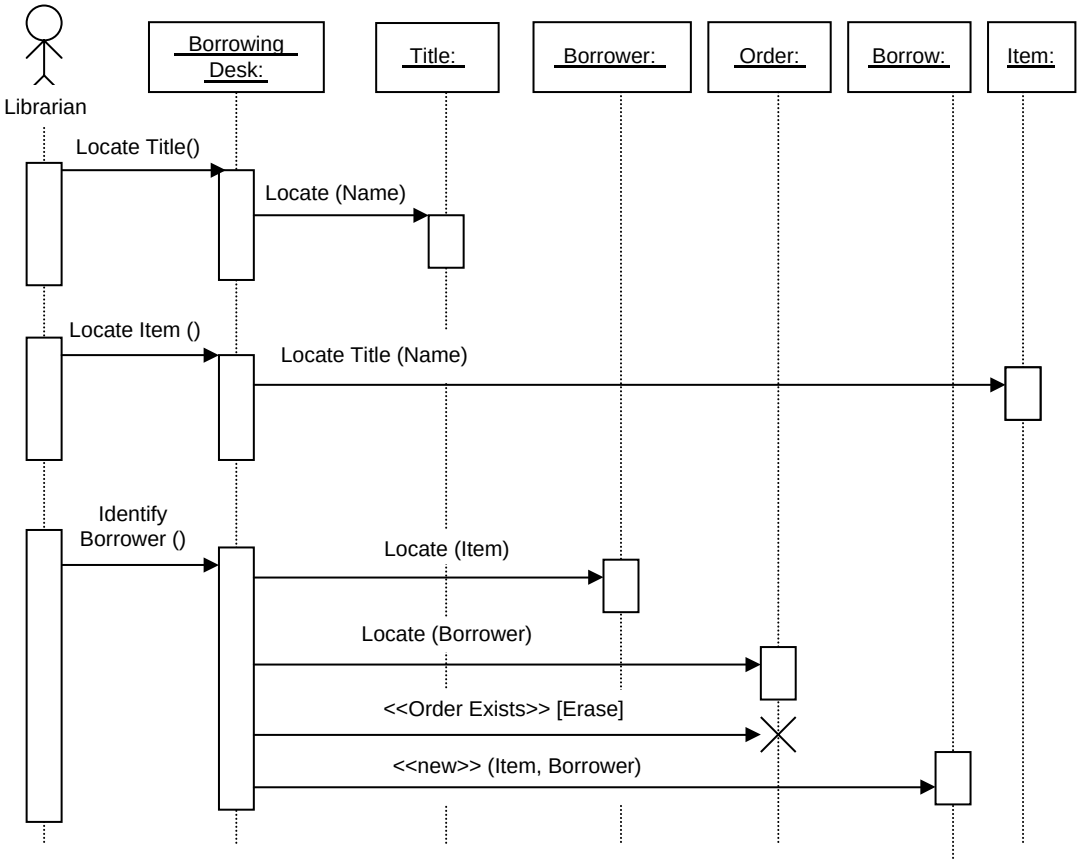

Fig. 1. Sequence Diagram - Library MIS.