In J. Kuljis, L. Baldwin & R. Scoble (Eds). Proc. PPIG 14 Pages 1-14

A Comparison of Empirical Study and Cognitive Dimensions Analysis in the Evaluation of UML Diagrams

Maria Kutar, Carol Britton and Trevor Barker
Department of Computer Science
University of Hertfordshire, UK
M.S.Kutar, C.Britton, T.1.Barker@herts.ac.uk

Keywords: POP-I.B. Choice of Methodology POP-II.A. Individual Differences POP-III.C. Cognitive Dimensions POP-III.D. Specification Languages POP-IV.A. Object-oriented Design

Abstract

It is important that all those who use representations of a system during the development process can clearly understand the representations that are used. Research has shown that structure plays an important role in whether a diagrammatic representation may be readily understood. In this paper we present the results of a study where two different approaches were taken to the evaluation of two notations which form part of the UML diagram toolkit: sequence diagrams and collaboration diagrams. First, a theoretical investigation was carried out using the cognitive dimensions framework. Second, an empirical study was carried out to investigate user understanding of such diagrams. The results of the two studies did not concur, with the theoretical approach supporting the original hypothesis that structure is an important factor in diagram comprehension, but the study providing no evidence to support this.

1. Introduction

The selection of appropriate tools for use in systems development can influence the success of the development process. Scenarios are commonly used during the requirements process both as a means of communication between users and developers, and to elicit and validate system requirements. Scenarios may be diagrammatically represented, and the Unified Modelling Language (UML) (Booch et al.99) provides two diagrams for the representation of scenarios, sequence and collaboration diagrams (known collectively as ‘interaction diagrams’). The two diagrams are semantically equivalent, and the simpler forms are isomorphic. The key difference between the two diagram types is the structure of the information that is represented.

It is recognised that the structure of information representations may have an impact on the ability of users to understand such representations (Britton 00). One way of examining this impact would be to carry out empirical studies of users understanding of representations using the different diagram types. An alternative approach would be to evaluate the structure of the diagrams using some analytical framework. One such framework is the Cognitive Dimensions Framework (Green 89) which is an evaluative framework for information structures. The research presented in this paper investigated the difference in user’s understanding for the two different types of interaction diagram using both approaches. Our hypothesis was that the nature of the structure of sequence diagrams would enhance readers ability to extract information as compared to sequence diagrams. Firstly, a cognitive dimensions analysis was conducted for each type of diagram. Secondly a study was carried out to investigate which type of diagram users were better able to extract information from. The results of the two approaches were compared, to establish whether the cognitive dimensions analyses were in line with the findings of the empirical study.

The rest of this paper is structured as follows:

- in section 2 we give an introduction to sequence and collaboration diagrams
- in section 3 we discuss the cognitive dimensions analysis of the two diagrams
- in section 4 our study of user understanding of the two diagram types is introduced, and the results summarised
- in section 5 we give our conclusions and identify directions for future work.

2. Introduction to Interaction Diagrams

The two most important models contained within the Unified Modeling Language (UML) are the use case model (which enables description of the tasks that must be performed by, or with the assistance of the system), and the class diagram which describes the classes that will be used to achieve this and the relationships between them (Pooley & Stevens, 1999). *Interaction Diagrams* are a further component of UML, providing a detailed record of the way in which objects interact to perform a task and thus providing a bridge between the use case model and the class diagram. Two different types of interaction diagram are provided by UML, both providing (almost) the same information. These are the *Collaboration diagram* and the *Sequence diagram*. The two diagrams are semantically equivalent, and many CASE tools generate one from the other (e.g. Rational Rose). In this paper we consider only the *instance* form of interaction diagrams which may be used to describe the interactions between objects in a particular *scenario*. A scenario may defined as representing "one instance of a use case, describing a particular sequence of events that may occur in trying to reach the use case goal" (Britton & Doake, 2000). Scenarios are recognised as playing an important role throughout the development process, and "can be used as a tool in requirements gathering, interface design and evaluation" (Bennett et al., 1999). In order to illustrate sequence and collaboration diagrams, we will use the example of a simple scenario for the use case 'read email'.

The scenario is shown below:

- The user inputs a request on the PC to read his/her email messages
- The user is asked for his / her password
- The user inputs the password
- The PC sends the email request and the password to the central computer
- The central computer checks that the password is valid
- The central computer sends the user's messages to the PC
- The PC displays the messages to the user

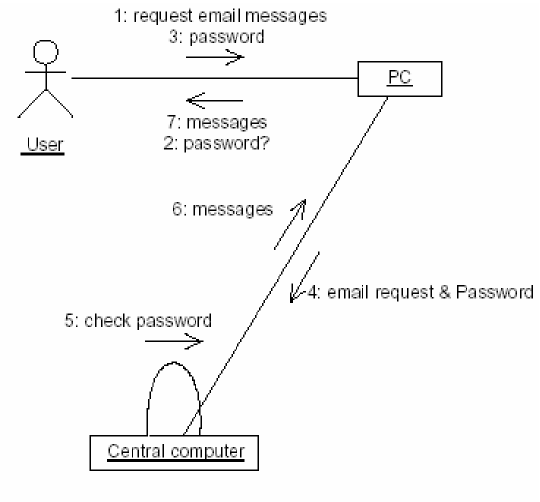

Figure 1: Collaboration diagram for the scenario 'reading an email message'.

The *Collaboration Diagram* is a representation of the objects which interact to perform a particular task and the interactions between those objects. The components of the diagram are objects, actors, links and interactions. Links show the relationships between the objects and actors involved in a particular scenario, and interactions describe the messages which pass between the components. Figure 1 shows a collaboration diagram for the scenario above, 'reading an email message'.

The rectangles represent *objects*, the stick man an *actor*, and the lines between the various components show the *links* between them. Text alongside a link describes the *interaction*. The collaboration diagram clearly shows the relationships between the various objects and their interactions.

The *Sequence diagram* is a representation of the same basic information that is contained within the collaboration diagram. Here however, the structure is changed to emphasise the dimension of time. (This is shown in the collaboration diagram via the numbers on the messages.) The sequence diagram for the scenario 'reading an email message' is shown in figure 2.

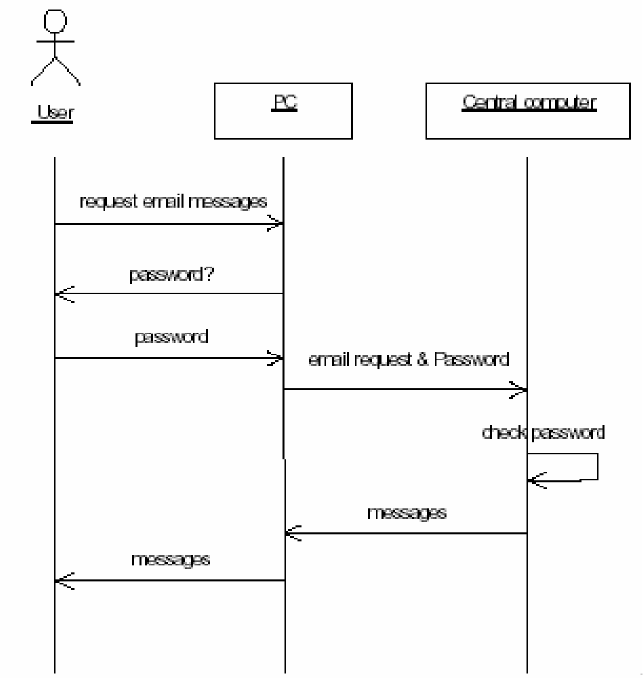

Figure 2: Sequence Diagram for scenario 'reading an email message'.

The objects and actors are shown at the top of the diagram, each with a dotted line below which represents their *lifeline*. An arrow from the lifeline of an object (or actor) to that of another object, represents the link, annotated with details of the interaction in the same way as in the collaboration diagram. It may be seen that the two diagrams contain the same objects, actors, links and interactions. The only difference between the two representations is that the structure of the same underlying information is changed.

The provision of two different diagrammatic structures to represent the same underlying information enables different interpretations of that information to be made more readily. The collaboration diagram is better at showing the links between objects, but the sequence diagram enables the order of interactions to be seen more clearly (Pooley & Stevens, 1999). This suggests that a choice must be made as to which diagram should be used in a given situation. Maciaszek (2001), for example, suggests that sequence diagrams are more effective in analysis models and collaboration diagrams in design models.

3. Cognitive Dimensions Analysis

It is generally accepted that the use of different languages for representation has an impact on the effectiveness with which a variety of tasks can be performed (Stenning & Oberlander, 1995). This is true especially in software system development, where the effect of the choice of language on successful system development has long been recognised (Green, 1989, 1991; Johnson, McCarthy & Wright, 1995; McCluskey et al., 1995; Modugno, Green & Myers, 1994; Roast, 1997). However, the relationship between languages, representations and the quality of the system development process is not fully understood. Little is currently known about what languages are likely to be most suitable for use in which contexts. The choice of languages for particular projects often reflects the experience or preferences of the development team more than an objective consideration of possible alternatives (McCluskey et al., 1995).

We believe that selection of notations would be more effective, were it influenced by consideration of whether a language is fit for a particular purpose, through an evaluation of that language. The difficulty of evaluating representations directly has been recognised by other authors. These include Stenning and Oberlander (1995) who note problems with an approach which emphasises "differences between token representations, rather than the differences of expressive power of the systems the tokens are drawn from".

3.1 Introduction to Cognitive Dimensions

In this section we consider the concept and purpose of cognitive dimensions and give a brief overview of the individual dimensions. (For a full guide to cognitive dimensions the reader is directed to (Green & Blackwell 1998)). The aim of the cognitive dimensions framework is to provide tools which may be used to evaluate the usability of information structures. They are ‘thinking tools’ rather than strict guidelines, with a focus on usability aspects which make learning and doing hard for mental, rather than physical reasons. In Thomas Green’s words, "When a train of thought is broken again and again by the need to find something out the hard way, it is difficult to piece thoughts together into inspirations; it is difficult enough even to finish a simple train of thought without making a mistake, simply because of having to get the information in some tedious and error-prone way." (Green 1989). Cognitive dimensions are aimed at the non-HCI specialist and therefore comprise a broad-brush approach rather than detailed guidelines.

The cognitive dimensions framework may be applied to both interactive artefacts such as word processors, and non-interactive artefacts such as music notation, and programming or specification languages. An artefact may be analysed and a usability profile derived which can assist in determining the artefact’s suitability for a particular task. It should be noted that the artefact is considered in conjunction with the environment in which it is to be used. We may think of the combination of an artefact and it’s environment as a ‘system’, and it is this combination, the ‘system’, to which the analysis is applied. Consequently, a single specification language, for example, may be considered in a number of different environments, each ‘system’ resulting in a different usability profile. This is a key feature of cognitive dimensions as rather than providing a generalised analysis, they may be used to evaluate an artefact’s suitability for a particular purpose. The ability to consider an information structure in the context in which it is used greatly enhances the usefulness of the dimensions.

Dimension	Description
Abstraction	Types / availability of abstraction mechanisms
Hidden Dependencies	Important links between entities not visible
Secondary Notation	Extra information in means other than formal syntax
Diffuseness	Verbosity of language
Premature Commitment	Constraints on the order of doing things
Viscosity	Resistance to change
Visibility	Ability to view components easily
Closeness of Mapping	Closeness of representations to domain
Consistency	Similar semantics are presented in a similar syntactic style
Error-Proneness	Notation invites mistakes
Hard Mental Operations	High demand on cognitive resources
Progressive Evaluation	Work-to-date can be checked at any time
Provisionality	Degree of Commitment to actions or marks
Role Expressiveness	The purpose of a component is readily inferred

Table 1: The Cognitive Dimensions

3.2 Summary of Cognitive Dimensions Analyses of Interaction Diagrams

The two different diagram types were analysed by the authors using the cognitive dimensions framework. Each author spent some time thinking about how each cognitive dimension applied to the different diagrams. The authors then discussed their findings for each dimension, examining how it may impact on development and reading activities. The results of our analysis presented below were agreed by all authors. The analysis examined the diagram where the requirements engineering process is the context of use. It is assumed that for this context of use the diagrams will be produced by practitioners familiar with the diagram type. Readers of the diagrams will include both systems development personnel as well as system end users who may be asked to validate information represented by the diagrams (and who may not necessarily be familiar with the mode of representation).

Dimension	**Sequence Diagram**	**Collaboration Diagram**
Abstraction	Objects / actors / events Stick man is a useful starting point	Objects / actors / events Emphasis on object relationships means the developer is better supported than readers (and leads to a need for transitions to be numbered). Stick man useful starting point
Hidden Dependencies	Can be difficult to establish relationships between objects Explicit ordering of messages via lifelines	Relationships, especially clusters are apparent Ordering of messages can be hard to ascertain
Secondary Notation	Temporal information is implicit in diagram structure Order of objects at top of diagram can affect readability	Little potential. Layout & ordering of messages can be problematic, but careful use can aid reading
Diffuseness		Can become unreadable quickly if terse language not used
Premature Commitment	Ok	Positioning of objects requires care
Viscosity	Ok	Numbering system means it is hard to change a message
Visibility	All component parts highly visible	Ordering difficult to ascertain. If diagram becomes cluttered with transitions visibility is reduced
Closeness of Mapping	Scenarios represent sequences of events. This is clearly identified in the sequence diagram	Closer to programmer than user because of emphasis on object links
Consistency	Ok	
Error-Proneness		Can be error-prone for readers
Hard Mental Operations	Using either diagram for the ‘wrong’ purpose will incur HMO’s. Make both representations available?	
Progressive Evaluation	Ok	
Provisionality	Ok	
Role Expressiveness	Stickman is role expressive	

Table 2: Summary of Cognitive Dimensions analyses

Overall our cognitive dimensions analysis supports the view of Maciaszek (01) that sequence diagrams are more effective in analysis models and collaboration diagrams in design models. This is illustrated by our findings for the dimensions of hidden dependencies, secondary notation, visibility and closeness of mapping as shown in table 2 above. The analysis therefore supports the view that system end-users will be better able to extract information from sequence rather than collaboration diagrams. Our hypothesis at the start of the empirical study was that participants in the study would perform more accurately with sequence diagrams.

4. Study

4.1 Description

The purpose of the study described in this paper was to investigate whether users showed greater accuracy in understanding the information contained in sequence or in collaboration diagrams. From the research described above, it was expected that sequence diagrams would result in more accurate user performance. Each of the 124 participants in the study was a first year undergraduate in Computer Science from either the University of Hertfordshire or Anglia Polytechnic University. The experience of the students ranged between having no previous experience with either diagram, to having a little experience of both diagram types. None of the participants claimed to be an expert with either type of diagram.

The study was carried out using a questionnaire, which was produced in four versions and answered anonymously by the participants. The four versions of the questionnaire were distributed randomly amongst the groups of participants. Each version contained six scenarios:

- Making an appointment to see the doctor
- Using a lift
- Driving into a car park
- Ordering a book on the Internet
- Using directory enquiries
- Using a cash machine

In versions 1 and 3 of the questionnaire, scenarios 1, 3, and 5 were represented as sequence diagrams and scenarios 2, 4 and 6 as collaboration diagrams; in versions 2 and 4 of the questionnaire the representations were reversed. In addition, the order of the scenarios in versions 1 and 4 was different from the order in versions 2 and 3.

Each diagram in the questionnaire had five multiple-choice questions relating to the information contained in it; these questions were to be answered by the participants after studying the diagram. Since most of the participants were unfamiliar with the Unified Modelling Language and with these diagrams, the diagrams were referred to in the questionnaire simply as 'Type 1' and 'Type 2'. Participants were also asked to state which of the diagram types they thought they would prefer to work with before answering the questions, and which they actually found easier to work with after answering the questions. At the end of the questionnaire session (which lasted approximately 30 minutes) the scripts were collected and marked; scores were collated and subjected to analysis.

4.2 Results

Table 3 below presents a summary of the data obtained in the study.

mean data obtained in the study (N=124)

Condition	Mean Score (SD)
Type 1 Diagrams	3.16 (0.1)
Type 2 Diagrams	3.18 (0.1)
Scenario 1	3.12 (0.1)
Type 1	3.03 (0.1)
Type 2	3.21 (0.1)

Scenario 2	3.67 (0.1)
Type 1	3.79 (0.1)
Type 2	3.54 (0.1)
Scenario 3	1.95 (0.1)
Type 1	1.85 (0.1)
Type 2	2.06 (0.1)
Scenario 4	3.73 (0.1)
Type 1	3.67 (0.1)
Type 2	3.77 (0.1)
Scenario 5	2.96 (0.1)
Type 1	2.87 (0.1)
Type 2	3.05 (0.1)
Scenario 6	3.58 (0.1)
Type 1	3.71 (0.1)
Type 2	3.44 (0.1)

Table 3 - Summary of the mean data obtained in the study (N=124)

An Analysis of Variance (ANOVA) for the data presented in table 1 was performed to determine the significance of any differences in the means shown there. The results of this analysis are shown in table 4 below.

Table 4

Analysis of Variance performed on the data shown in table 1

Source	df	F	Signif.
Scenario	5	53.01	<0.001
Type	1	0.125	0.72
Scenario * Type	5	1.51	0.19

Table 4 Analysis of Variance performed on the data shown in table

Table 4 shows a significant difference in mean scores (p<0.001) due to the effect of the scenario on performance in the test. There was no significant difference observed due to the effect of diagram type (p=0.72).

These results of this analysis are interpreted as follows:

- There was no significant difference between user performance on sequence or collaboration diagrams
- There was a significant difference in mean scores due to the effect of the scenario on performance

5. Comparison of Findings

The hypothesis at the start of this research was that users should be more effective at extracting information from sequence diagrams than from collaboration diagrams. This belief stemmed from the difference in structure of the two diagram types.

The first stage of the research presented in this paper, the cognitive dimensions analyses of the two diagrams, supported our hypothesis. The indication was that properties of sequence diagrams relating to structure (illustrated by the findings for the dimensions of hidden dependencies, secondary notation, visibility and closeness of mapping as shown in table 2 above), were sufficiently different to those of collaboration diagrams that they would cause a significant difference in user performance across the two diagram types.

The second stage of the research yielded results that diverged from our initial hypothesis. Analysis of the data collected in the study showed that there was no significant difference in user performance across the two diagram types. The results of this study clearly do not conform to the findings of our cognitive dimensions analyses.

5.1 Discussion

We suggest that there may be a number of reasons for the disparity in findings between the two sections of our research. Firstly, although no significant difference in user performance was found across the diagram types, there was a significant difference across different scenarios. This suggests that users' knowledge of and familiarity with the domain may contribute to their ability to extract information to a greater extent than the structure of that representation. Secondly, it may be possible that the increased difficulty of following the structure of the collaboration diagrams may force users to work harder, thus increasing their level of efficiency with this diagram type to a level on a par with that of sequence diagrams. This observation is in line with the findings of Purchase et al. (2001), who studied variations of UML class diagrams and found that, where participants felt less at ease with a notation, they appeared to be more diligent in working with it, resulting in better performance.

Consideration must also be given to the study design. It is recognised that the sample is a homogenous group; as first year Computer Science students they are mostly males in the age range 18 – 24, and all subjects had either no experience or only a little previous experience with these diagrams. However, the students came from two different institutions, and the sample was of a sufficient size (124 subjects) that we are confident of the validity and reliability of the results.

6. Conclusions

We have investigated whether there is a difference between a user’s ability to extract information from sequence and collaboration diagrams which form part of the UML modelling toolkit. The two diagrams are semantically equivalent but differ in their structural representation of information. We examined the diagrams using both a theoretical approach (analysis using the cognitive dimensions framework) and a experimental approach (carrying out a study of users of the diagram). Our hypothesis was that the structure of sequence diagrams, which have a clear direction and explicitly show the sequential ordering of events, would be more easily understood by readers.

The hypothesis was supported by the theoretical investigation, with analysis under several of the cognitive dimensions indicating that the difference in structure would provide a significant difference in user understanding. However, the second investigation did not yield the same result; here there was no significant difference in performance according to diagram type.

We believe that the difference in findings between the two investigations may be caused by any of the following:

- Additional effort from users on tasks they perceive as more difficult
- User preference for a particular type of diagram
- Scenario effect
- Problems inherent in the CD’s framework
- Problems inherent in the design of the study

Further research is needed to investigate the disparity between the findings of the two investigations. Further studies are being carried out where users are familiar with the scenarios represented by the diagrams, in an attempt to overcome the ‘scenario effect’. In addition, further comparison of findings from theoretical and experimental investigations are planned, using a different notation.

References

Bennett, S., McRobb, S. & Farmer, R. (1999), *Object-Oriented Systems Analysis and Design Using UM*L, McGraw-Hill.

Booch, G., Jacobson, I. and Rumbaugh, J. (1999) *The unified modelling language user guide* Addison-Wesley.

Britton, C. (2000), Intelligible Specifications for Interactive Software Systems, PhD thesis, University of Hertfordshire.

Britton, C. & Doake, J. (2000), Object-Oriented System Development: A Gentle Introduction, McGraw-Hill.

Green, T. 1989. *Cognitive Dimensions of Notations*. *In:* Sutcliffe, A, & Macaulay, L (eds), *People and Computers V, Proceedings of HCI’89*. CUP.

Green, T. (1991). Describing information artefacts with cognitive dimensions and structure maps. In D. Diaper, & N. Hammond (Eds.), *People and Computers VI, Proceedings of HCI’91*. Cambridge University Press.

Green T R G and Blackwell, A F (1998) A tutorial on cognitive dimensions. Available on-line at: http://www.ndirect.co.uk/~thomas.green/workStuff/Papers/index.html

Johnson, C., McCarthy, J. & Wright, P. (1995). Using a formal language to support natural language in accident reports. *Ergonomics, 38 (6).*

Maciaszek, L. (2001), Requirements Analysis and System Design. Developing Information Systems with UML, Addison Wesley.

McCluskey, T., Porteous, J., Naik, Y., Taylor, C. & Jones, S. (1995). A requirements capture method and its use in an air traffic control application*. Software Practice and Experience, 25 (1).*

Modugno, F., Green T. & Myers B. (1994). Visual programming in a visual domain: A case study of cognitive dimensions. In *People and Computers IX, Proceedings of HCI'94*. Cambridge University Press.

Pooley, R. & Stevens, P. (1999), *Using UM*L, Addison Wesley.

Purchase, H.C., Colpoys, L., McGill, M., Carrington, D. and Britton, C. (2001) UML class diagram variations: an empirical study of comprehension. Presented at the Australian Information Visualisation conference, Sydney, 3 – 4 December 2001

Roast, C. (1997). Formally comparing and informing notation design. In H.Thimblely, B. O'Conaill & P. Thomas (Eds.), *People and Computers XII, Proceedings of HCI'97.* Springer.

Stenning, K. & Oberlander, J. (1995). A cognitive theory of graphical and linguistic reasoning: Logic and implementation. *Cognitive Science, 19*, 97-140.