Evaluating Using Animation to Improve Understanding of Sequence Diagrams

Elizabeth Burd, Dawn Overy, Ady Wheetman

The Research Institute in Software Evolution
University of Durham
South Road
Durham, DH1 3LE, UK

Abstract

This paper describes an experiment whereby the benefit of using animation to improve the comprehensibility of UML sequence diagrams is assessed. The paper hypothesizes that through animation the control flow of sequence diagram will become more evident. The development a system that seeks to enable stakeholders to better interpret UML modeling behaviour is described. This system aims to provide dynamic visualization through the use of animation techniques. A study to evaluate the extent to which the animation of a diagram can aid its interpretation is then described. The results of the study show that the animation system did improve the comprehensibility of the sequence diagram control flow thus improving the comprehensibility when compared to the comprehensibility of a traditional static representation. Finally, this paper discusses the reasoning for these results.

1. Introduction

Despite the massive investment that companies place in IT and the recent growth in performance in both hardware and software, the Standish Group reports [1] that only sixteen per cent of all projects are successful. Over a half are impaired while almost a third are cancelled. These impairments manifest themselves in many different ways. Projects are often delivered late and may fall short of established quality standards. The product may be unreliable due to insufficient testing, contain redundant functionality or even be incomplete. Such quality comprises maintainability. From the point of view of the developers, it is common for the product to have exceeded its budget. When put into a global business context, it is essential that issues relating to the performance of the IT industry are addressed.

It is hard to identify one major cause of the crisis. However, work on the problem has led to the recognition of a subset of crucial factors. The early stages of a project's lifecycle are crucial to its success. At each stage communication between developers and other stakeholders is undertaken. Such communication must be carried out successfully in order to achieve a valid and verified set of requirements. It is essential for different stakeholders to convey system attributes in an objective and unambiguous fashion, despite sometimes using different languages. The application of modelling to represent components of the project provides a way to overcome such communication difficulties.

A model is an abstraction or higher level description of an entity or system. Modelling therefore allows the design of a system to be considered and visualised early on. This can be an enlightening process and can present advantages and disadvantages of a system proposed design prior to the implementation stage. The shear size of todays systems far exceeds the understanding of any one person. The use of modelling is used to provide an interpretable abstraction.

The need for system modeling was addressed by Rumbaugh, Jacobson and Booch when they defined the Unified Modeling Language (UML). It is a "general-purpose modeling language that is used to specify, visualize, construct and document the artifacts of a software system" [2]. It is not a method advocating a step-by-step process but a toolkit to allow the description of a system. The semantics and notations inherent in models are exploited in UML to allow static and dynamic views that depict a system. One of the goals of UML is to allow the visualization of systems to enable interpretation and understanding by a wide variety of people. Such visualization is key to allow stakeholders to communicate in the same language and

Proceedings of the 10 th International Workshop on Program Comprehension (IWPC'02)
1092-8138/02 $17.00 © 2002 **IEEE**

developers to create a product that is exactly what the customer requires.

Visualisation is the graphical representation of an entity to convey the semantics of that entity to the viewer. As systems become more complex, dynamic visualisation or animation may be a solution of combat this problem.

Animation has the potential to address many of these issues raised through improving the levels of comprehension to allow requirements verification by technical and non-technical staff. A system that can interactively model and animate requirements could be used throughout the requirements engineering process by a number of stakeholders. Thus the lack of customer involvement can be addressed directly. Animation of a requirements diagram such as a UML sequence diagram can aid the stakeholders interpretation of the system and give them a greater understanding of the product proposed to be developed.

2. Background

Visualisation of a model is one approach to aid its understanding. The extent to which different types of visualisation effect the assimilation process is unclear and much work has been focussed in this area. Eloi [3] suggests that people carry underlying assumptions in their head, which when made explicit through model visualisation can be used as an explicit rule of inference or shared with others. He also suggests that participating in several visualisation sessions encourages the user to construct a dynamic mental model based on the animation itself. This hypothesis is corroborated by Letovsky's [4]. The visualisation system, combined with the existing knowledge base and both handled by the assimilation process drives the evolution of the user's mental model. Norman draws the conclusion that a person's ability to run their mental model are severely limited, [5]. On this, Eloi comments that visualisation support provides insight into a conceptual model. He suggests that participating in a visualization session can resolve differences between their mental model and what is perceived by the visualization model" i.e. visualization support facilitates cognition. Work by Byrne et al [6] suggests that dynamic visualization provides more opportunity for the participant to make and test ad-hoc spontaneous mental predictions regarding the behaviour of the system. Making and testing predictions are a form of active learning, which supports cognition more than passive learning.

This paper describes an experiment whereby the benefit of using animation to improve the comprehensibility of UML diagrams is assessed. The paper discusses the development a system that seeks to enable stakeholders to better interpret UML modeling behaviour so that they are more able to communicate and address problems. The system aims to allow dynamic visualization through the use of animation techniques. A study to evaluate the extent to which the animation of a diagram can aid its interpretation is then described and the results of the study are presented and discussed.

3. Approach

It is hypothesized that through animation the dynamic nature of UML diagrams can be made more explicit. This section describes the process adopted to test this hypothesis.

In order to investigate the benefit of animation to improve the understandability of the dynamic UML diagrams three main activities were performed. These were:

1. Identify requirements for the animation system
2. Implement a system to animate sequence diagrams
3. Compose an experiment to investigate if benefits through animation are achieved.

Each of these activities will now been described in further detail.

3.1 Identify Requirements for the animation system

The initial consideration for aiding the comprehension process through using animation was to determine what aspects of the UML to animate. Clearly the benefits of

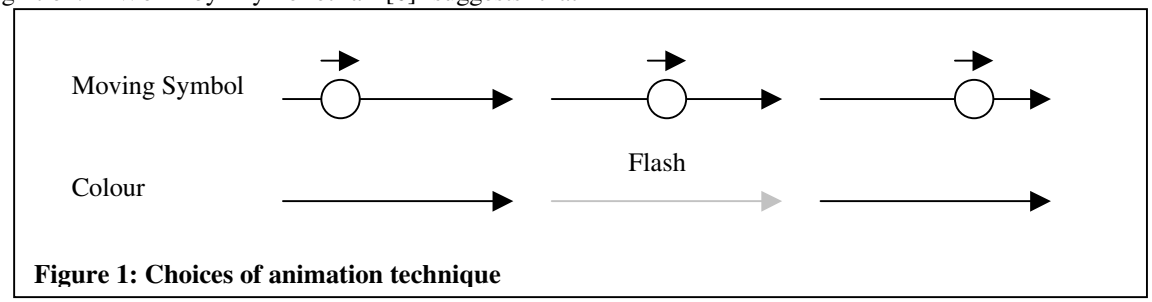

Figure 1: Choices of animation technique

animation could best be utilized through the representation of its dynamic aspects. Of these dynamic aspects it is often sequence and collaboration diagrams that are considered the hardest to comprehend and therefore could receive the greatest assistance towards improving comprehension. Since the sequence and collaboration are semantically equivalent then the selection between these two model types is arbitrary. Sequence diagrams were selected due to their more clearly defined layout properties.

In order to enhance the comprehensibility of the sequence diagram it is initially necessary to decide which concepts should be the focus of animation and would best support understanding through animation. The decision included considering object creation/deletion and message passing objects via activations or messages. To achieve the animation moving symbols could pass through the diagram to indicate the control flow, similar to Process Weaver [7], which uses a CASE tool to animate Petri Nets. Alternatively a more primitive animation effect is the use of colour to highlight the control flow. Examples of these two approaches are highlighted in Figure 1.

The main representational property of sequence diagrams is that of control flow. Therefore animation properties are best utilized though the animation of those aspects of the diagram that represent this flow of control. For this reason it was decided to animate the process of messages passing between the objects.

The choice of animation effect was then considered. If the animation was to represent control flow between objects then the best methods for highlighting ones attention to the change of control between objects needs to be found. For a sequence diagram having a symbol moving across a message line using Process Weaver techniques does not have any additional meaning within a sequence chart. It is simply the fact that the message has been sent there is no semantic meaning, for instance, in the speed of the message passing. Therefore, the simpler animation technique where the entire message passing line is temporarily coloured and flashes once is adopted.

3.2 Implement a system to animate sequence diagrams

A simple implementation has been developed. The sequence animation system is written in Java and takes a single input file that describes the sequence diagram to be animated. The animation occurs on the basis of the order of the input file. This therefore means that data format of the input file is critical, however, if the system proves successful then further enhancements to improve the usability of the system can later be made.

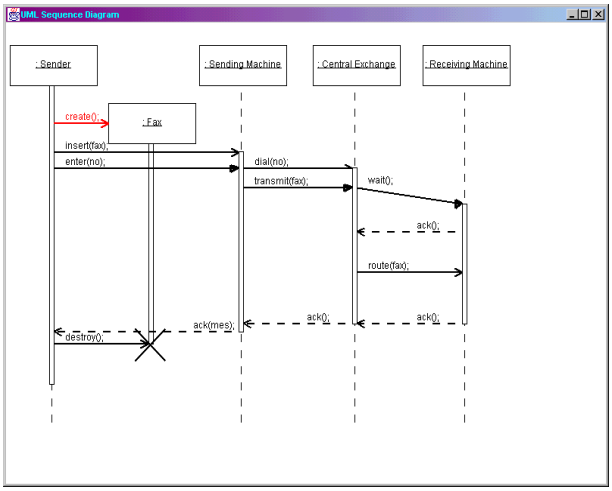

Figure 2: A simple sequence diagram in the animator system

A screen shot of a typical sequence diagram that could be animated is shown in Figure 2. This figure shows the object in square boxes at the top of the diagram and the lifelines of each object are represented by the lines down from each object. The lifelines from top to bottom of the page represents passing time. The arrow between the lifelines represents the message passing between objects and is the aspect of the diagram that is animated. For further details on sequence diagram specification and interpretation please refer to [2].

3.3 Compose experiment to investigate if benefits through animation are achieved

In order to investigate if the use of animation improves the comprehensibility of sequence diagrams then it is necessary to initially select a domain and scenario so an experiment to be conducted. It was decided that a fairly simple and familiar scenario should be adopted and thus the scenario sending a fax was used.

In order to investigate comprehensibility at different levels it was decided to investigate the benefits of animation for simple UML sequence diagrams and compare this to the benefit to comprehensibility for more complex examples. The modification of the scenario was then necessary that portrayed different levels of abstraction for the simple and complex sequence diagrams. A decision was made to have the scenario "sending a fax" as reusable components could be extended to provide sufficient combinations to model both examples. The simple sequence diagram was primarily concerned with a single use case, therefore the

decision to portray the best-case scenario for sending a fax was justified in terms of usability. As a result of this decision an instance sequence diagram was designed to represent a fax that could be successfully sent at the first attempt. The complex sequence diagram was developed to portray multiple paths through the system. Due to the numerous possibilities of unsuccessful path combinations that can arise when sending a fax, only one was represented to improve *usability* and visual simplicity. The conditional behaviour of the generic diagram depicted the phone line when it was free or busy. In addition to the successful routing of the fax a report was generated recursively. In UML terminology is more complex diagram would be referred to as a generic sequence diagram.

Once the scenario had been chosen and written in UML then an experiment and questionnaire was produced to test the hypothesis. The questionnaire was designed to be discretionary and anonymous and its objective was to compare the static (graphics) and animated versions of the UML sequence diagrams to determine user's interpretations before and after animation. The questionnaire comprised of ten questions and four diagrams. Two of these diagrams were supported by an animated example. Users were instructed to annotate the diagrams with their interpretation of the control flow for the static and animated versions consecutively.

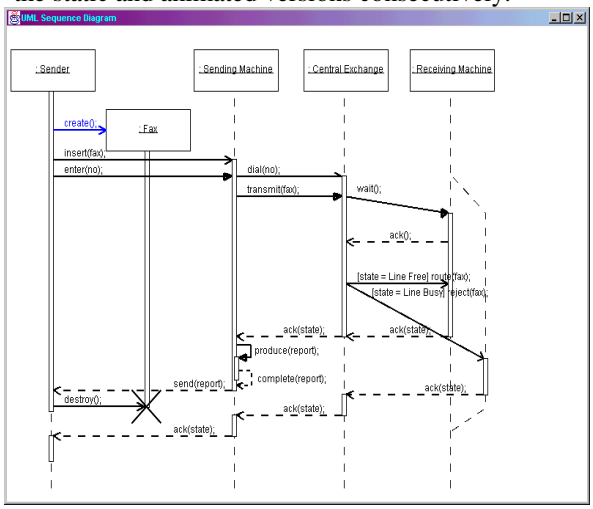

Figure 3: A complex sequence diagram in the animator system

The experiment and questionnaire compared the understanding of two sets of sequence diagrams. The first of these sequence diagrams was fairly simple (see Fig 2) the second more complex (see Fig 3). By comparing the static and animated interpretations, the benefit of animation and participants comprehension of the control flow was determined. The use of a complex and simple sequence diagram allowed investigations of how complexity effects comprehension and how this understanding is helped or hindered through the provision of animation. Both simple and complex diagrams represent the same scenario but in the case of the complex sequence diagram additional conditions are shown.

In addition, the questionnaire also offered opportunities for participants to signify their animation preferences, provide suggested improvements and comments and suggest potential UML models for animation. Participant's reactions were observed as they ran the animated deliverables to clarify if the control flow was obvious from the animation.

An adequate sample was important to ensure fair representations of participant's views were obtained. Therefore it was decided that people with different levels of expertise in UML would act as the participants. Forty people were distributed with a questionnaire; fifteen had no prior knowledge of UML, twenty had up-to-one year's knowledge, four had two-to-four year's knowledge and one had five or more, as identified from the questionnaires. Those with one year's or less experience of UML were students at the University of Durham, the others were experts working in industry. People with no prior knowledge of UML were chosen randomly from outside the Computer Science domain. A larger sample was obtained for up-to-one year's knowledge as the second and third year Computer Science syllabuses incorporate UML in Durham. Obtaining expert knowledge proved to be more difficult, due to the experts time commitments. By utilising a UML devoted mailing list known as the Object Technology User's Group a minority of experts offered their time to complete the designated questionnaire. The mailing list provided a source of experts who had common interests in object technology, which was ideal for evaluation purposes. A different evaluation technique was adopted for the latter evaluation, as it was not possible to observe the experts running the programs. Consequently class files for were sent via email with instructions outlined in the questionnaire.

Although slight variation in techniques were adopted it was fundamental to obtain such feedback on this work to establish the benefits for comprehension for different levels of expertise. The reasoning behind the sample was to focus on participants who would benefit most from the animation such as people with no prior knowledge or minimal experience of UML. Expert knowledge would then provide an industrial perspective and realisation of the effectiveness of the animation

4. Results

After collating the completed questionnaires the results were analyzed. These are now presented within the following sections. Initially the simple sequence diagram is investigate followed by the more complex analysis. Finally, some additional results obtained from the questionnaire are presented.

4.1 Simple UML Sequence Diagram

By analysing the annotated diagrams of the questionnaire, it was determined whether the correct interpretation of control flow had been established. Table 1 represents the results for the corresponding simple sequence diagram interpretations.

Diagram investigated (Fig. 2)	Correct interpretation	Incorrect interpretation
Static	21	19
Animated	36	4

Table 1: Summary of analysis results of simple sequence diagram

The results highlight that mistakes were made with both the static and dynamic versions. However, in the case of the animated version 90% of the participants interpreted the control flow correctly.

4.2 Complex UML Sequence Diagram

Table 2 represents the results for the corresponding complex sequence diagram interpretations.

Diagram investigated (Fig. 3)	Correct interpretation	Incorrect interpretations
Static	0	40
Animated	25	15

Table 2: Summary of analysis results of complex sequence diagram

The results from individual questionnaires showed a trend that participants did not expect a second control flow to represent the conditional branch. Twelve participants attempted to account for the branching by annotating both conditional branches simultaneously but did not account for recursive behaviour and six participants interpreted the control flow to be consecutive. Such interpretations did not match the predefined criteria in the study to qualify as a correct interpretation.

4.3 Other Results from Survey

The survey also queried which type of representation the participants preferred; that is the static or dynamic (animated) representations of the sequence diagrams. In the questionnaire participants were asked if the control flow was more obvious in the static or animated version to determine the significance of the animation. Table 3 identifies all preferences were for the animated version for both simple and complex diagram. However, individual questionnaires identified that two participants made additional comments that neither version was significantly more prominent, for the purpose of interpreting the simple diagram.

Question Description	Static	Animated
In which simple sequence diagram was the control flow more obvious.	0	40
In which complex sequence diagram was the control flow more obvious.	0	40

Table 3: Summary of participant preferences for type of sequence diagram.

5. Evaluation

Results from the study illustrated an 80% decrease in misinterpretations of the control flow when animation was utilized in the simple sequence diagram. Thus from this figure it can be argued that for the simple sequence diagram the benefit of animation was not significant since the level of misinterpretations was reduced to only 10%. However, this needs to be reviewed in light of the relative simplicity of the sequence diagram being interpreted. For such a simple diagram a much higher proportion of mistakes than one would expect were made for the interpretation of control flow. For the simple diagram the misinterpretations for the static diagram was 48% of the participants. This seems to indicate, certainly from those new to the UML notation, sequence diagrams are hard to comprehend. For software development this does imply significant

problems, as the original specification misinterpretations may result in incorrect implementations. In addition, the misinterpretations may additionally lead to problems during maintenance. If the UML sequence diagrams are used to interpret the implementation order for maintenance then the misinterpretations may result in the addition of further errors in the code. In addition, with technology now available to reverse engineeer UML specifications from existing code [8] it is more likely in future that these types of interpretations will be relied upon the implement change requests.

For the complex sequence diagram a 62% decrease was recorded for the control flow misinterpretations of the animated diagram compared to the static representation. There are a number of interesting features represented within this figure. The first of these is that for the static diagram all questionnaire respondents misinterpreted the control flow. This therefore seems to indicate that the difficulties associated with interpretation of control flow from these diagrams is not restricted to those with relatively little experience (one year or less) of the UML. In this case those with greater experience still made mistakes in its interpretation. The second important aspect of the results of the analysis of the complex sequence diagram was that there was still a relatively high proportion of mistakes made (38%) even with animation. Thus this high level of errors does identify issues of using sequence diagrams as means to develop or interpret software applications especially with using staff with relatively little experience of the UML.

Most importantly however this experiment would seem to indicate that through the use of animation dramatic reductions in the misinterpretation of the control flow occur. This would therefore seem to imply that by using animation to aid the identification of the focal point of the diagram, it has the potential to some extent, alleviate the diagram interpretation complexity.

The unambiguous result of this experiment and survey is that participants liked using the animation application as a means of interpreting sequence diagrams. The participants felt that the animator gave life to the models and as such felt that it improved their comprehension of control flow. The results of the experiment seem to concur with this observation.

While the results of the study seem to indicate a strong positive correlation that animation does significantly help with interpretation of the control flow of sequence diagrams there are a number of other elements that were hard to exclude from the study that may have lead to the positive results. The first of the elements is that of the potential practice effect. In order to evaluate our hypothesis is was necessary to fix the order of presentation of diagrams; first the static representation was shown followed by the animated representation. This may have meant that the second viewing of the diagram resulted in the improved understanding. However, to minimize this problem participants were allowed as long as they thought necessary to interpret the static diagram. The other important element was that of the novelty of the application. The novelty aspect of the animator may have encouraged participants to concentrate harder on the control flow and therefore have resulted in the improved comprehension. However, this novelty element potentially improving concentration is obviously a point in the system's favour.

6. Conclusions

This paper has conducted an experiment to investigate if animation can aid the comprehension process. Specifically this experiment examined if comprehension of control flow for sequence diagrams became more explicit through the use of animation. That is that participants would make fewer errors in interpreting the order and meaning of the control flow using an animated rather than a static representation. The experiment additionally investigated if the complexity of the sequence diagram effected the results obtained.

The results of the study showed that UML diagrams were hard to comprehend and participants made a number of errors in the interpretation of the control flow. However, the study in each trial showed an improvement in comprehension (fewer mistakes made) in the animated diagram representation over that of the static representation. The trials investigating the varying levels of complexity revealed that as the complexity of the diagram so correspondingly did the number of errors. However, in the case of the animated diagram the number of errors were significantly fewer than for the static diagram. The evaluation discussed some reasons why these findings were achieved.
In conclusion the animation effect proved a popular addition to aid comprehension and overall far fewer mistakes were made when this facility was available to users.

Further work needs now to be conducted to investigate if these results apply to more experienced users (5 years plus). However it seems from an educational viewpoint that animation does benefit those with little experience of the UML.

References

1 Standish Group (1995) "Chaos", http://www.standishgroup.com/visitor/chaos.htm. Last visited April 24 2000.

2 Rumbaugh, James, Jacobson, Ivor & Booch, Grady (1999) "The Unified Modelling Language Reference Manual", Addison-Wesley.

3 Eloi, Regis (1998) "Automated Support for Animating KISS Workflow Models to Enhance Client Understanding", Thesis for Degree of Master of Science by Research in the University of Hull, submitted September 1998.

4 Letovsky, S. & Soloway, E. (1986) "Delocalized Plans and Program Comprehension" in IEEE Software, May 1996, Vol. 19, No. 3, pages 41-48.

5 Norman, D.A. (1983) "Some observations on mental models" in Gentner, Dedre & Stevens, Albert L. (eds) (1983) "Mental Models", Lawrence Erlbaum Associated, Hillsdale, NJ, USA.

6 Byrne, Michael D., Catrambone, Richard, Stasko, John T. (1996) "Do Algorithm Animations Aid Learning?", Graphics, Visualization and Usability Center, Georgia Institute of Technology, Atlanta, GA, Technical Report GIT-GVU-96-18, May 1996.

7 Fernström, Christer , (1993) "Process Weaver. Adding Process Support to UNIX" in IEEE Press, September 1993.

8 DiLucca G.A., Fadoline A.R., De Carlini U., (2000) "Recovering Class Diagrams from Data-Intensive Legacy Systems" in International Conference on Software Maintenance, IEEE Press, October 2000.