Experimenting with Layout and Notation in UML Diagrams

D. Carrington, L. Colpoys, M. McGill and H. Purchase
School of Information Technology and Electrical Engineering, The University of Queensland
St Lucia, QLD 4072, Australia
{davec, lindac, mmcgill}@itee.uq.edu.au, hcp@dcs.gla.ac.uk

Abstract

This paper summarises an empirical research project that is investigating the impact of UML diagram notation and layout characteristics. The goal is to improve the effectiveness of these diagrams by identifying factors that assist human comprehension.

1. Introduction

This research aims to improve the effectiveness of software engineering diagrams from the human comprehension perspective. This can only be done through empirical research and has not been attempted previously. We are investigating the effective layout of diagrams in the Unified Modeling Language (UML), which has become the accepted notation for object-oriented analysis and design diagrams. In summary, this research is aiming to:

- Investigate the importance of layout characteristics and notational variations for UML diagrams through controlled experiments with the objective of identifying and quantifying characteristics that assist human comprehension of such diagrams.
- Propose empirically based principles for algorithms used in laying out UML diagrams.
- Develop a new algorithm that exploits this new knowledge of important layout characteristics.
- Validate the results by further experiments using the new algorithm in a modified UML tool.

UML diagram layout can benefit from existing and ongoing research into graph drawing [1]. Graph drawing has an active community of researchers working on automatic graph layout algorithms. These algorithms tend to consider only syntactic graph structure, rather than the semantics of the application domain. For example, they do not consider that different arcs may represent different relationships (e.g., inheritance or association), and do not distinguish between them when determining layout. Where UML tools do provide automatic layout, they generally rely on classical graph drawing algorithms such as spring layout [2] or Sugiyama [8], which is not necessarily appropriate for UML. One of the major reasons for the limited integration is the algorithmic and computational focus of much graph drawing research, with little or no experimental evaluation to assess the effectiveness of layout algorithms from a human perspective.

Our research is focusing on UML class and collaboration diagrams. Class diagrams provide a static or structural view of the classes and their relationships in a software system. In contrast, collaboration diagrams provide a dynamic or behavioural view of objects and their interactions. Both diagram types offer considerable flexibility in their layout. We have investigated both user preference and user performance for a variety of layout aesthetics and UML notational variations with these two types of diagrams.

2. Experimental Methods

Our experiments used second and third year Information Technology students who had familiarity with UML and similar graphical notations. At least 30 subjects were used in each experiment in order to provide statistical integrity.

We tested user preference for the following aesthetic and notational variations: bends, edge crossings, width of layout, font type, text direction, orthogonality, inheritance (class diagrams), number of arcs (collaboration diagrams) and directional indicators. Subjects were given two diagrams for each aesthetic/notation and were asked which one they preferred and why [4].

For all UML experiments, subjects were given preparatory materials to ensure familiarity with UML syntax and semantics, and they were shown the alternative, equivalent notations where necessary. The experimental task for the performance experiments was to match a given natural language specification to the UML diagrams displayed on screen. Subjects had to respond either yes or no depending on whether they believed the diagram accurately represented the specification. The
measures of performance were the correctness of the response and the time to respond, where a failure to respond within a time limit was considered an incorrect response [5].

For the notation experiments, a control diagram was created in several differing layouts to avoid subjects using visual pattern matching. The control diagrams then had all the notational variants applied individually, creating five correct diagrams for each layout as well as the control. Errors relating to the notational variants were applied to the control diagram and its variant for some of the layouts, producing a set of incorrect diagrams.

The layout experiments identified and tested the following aesthetics: edge bends (class diagrams only) [6], node distribution, edge lengths, edge length variation, direction of flow, orthogonality, and symmetry. The experimental diagrams included three diagrams for each aesthetic, except the flow aesthetic in the collaboration diagrams, which had four directions [7]. For each of the other aesthetics however, a diagram displaying a low, middle and high presence of that aesthetic was created. Aesthetic variations were applied independently through perceptual experiments to control other aesthetics that could cause confounding factors in a diagram set.

3. Results

Not all of our experiments produced significant results. We summarise those results where the significance was 0.05 or less (that is, less than a 5% chance there was no real difference in the results being compared). The choice of statistical test used in the analysis depended on the experiment [3].

In user preference experiments, the results show that subjects significantly prefer fewer crosses and bends; horizontal text; compact diagrams; orthogonality; consistent use of fonts; for both class and collaboration diagrams. Specifically for class diagrams, subjects prefer joined inheritance relationships and directional indicators on association relationships. For collaboration diagrams, subjects prefer longer arrows on messages and single, directed edges [4].

The performance experiments for UML class diagrams show that inheritance relationships flowing upward produce higher accuracy for identifying correct diagrams, but the opposite is true when identifying errors. In addition, errors are identified faster for downward flowing inheritance. Diagrams with separate inheritance arcs produce significantly more accurate results for identifying errors than diagrams with joined inheritance arcs. Using association classes improves accuracy for identifying errors compared to normal association relationships. Explicit representation of cardinalities gives better accuracy than abbreviated cardinalities when identifying correct diagrams, but is significantly slower [5].

Experiments with UML class diagram layout show that many bends gives slower performance than a small number of bent edges or none at all [6].

Performance data from the notation experiment on UML collaboration diagrams show that the simple numbering scheme is faster for identifying correct diagrams than hierarchical numbering [7].

Results from the experiment on UML collaboration diagram layout show that a medium amount of edge length variation gives slower performance than minimal edge length variation, and a large amount of edge length variation gives less accurate performance than either constant edge lengths or a medium amount of edge length variation.

4. References


