An experimental study into the default reading of constraint diagrams.

Andrew Fish
University of Brighton
Andrew.fish@brighton.ac.uk

Judith Masthoff
University of Aberdeen
jmasthof@csd.abdn.ac.uk

Introduction

Constraint diagrams [8] are a complex diagrammatic notation designed to express logical statements especially for use in software specification and reasoning. Not surprisingly, since this is an expressive language, there are some difficulties in reading the semantics of a diagram unambiguously. Some extra annotations (in the form of a reading tree) disambiguate the diagrams [1, 2]. However, this extra requirement (of drawing a reading tree) places a burden on the user. An attempt to remove the need for such a reading tree (or perhaps to automatically generate a reading tree, which could be altered by a user if they wished to) has been given via an algorithm to generate a default reading [4] from the diagram. This algorithm is based on a number of principles – most of which are properties of the diagram.

We wish to know whether these principles are intuitive and whether the default reading reflects a good proportion of users' intuitions, and we have performed a user-based study to test this. This report summarizes this study, for more detail see [5].

Reasons for the test

If we could obtain a default reading that is intuitive then we could:

- impose the default meaning – this reduces the complexity of a diagram (by removing the need for reading trees), but is likely to reduce expressiveness and will reduce the "ease of expression" (a diagram usually can be given different meanings via different reading trees), or
- generate the default tree for a diagram automatically, but allow users to specify a different tree if they wish to. The user's attention could also be drawn to points in which the diagram's meaning deviates from standard intuition (highlighting a spider that is an unexpected starting point of a reading for example).

If some of the principles are deemed intuitive, but not the overall default reading, and we could estimate the relative weighting of these principles (which ones take precedence over others), then we could redesign the default reading to reflect this, making it more intuitive.

If no principle can be identified as being intuitive then either:

- users could be taught a default reading (i.e. we could choose a default reading – but how to do this sensibly without guidance due to user preference?), or
- the idea of default could be abandoned and one could assert that the complexity associated with drawing a reading tree is necessary if one wishes to avoid ambiguity whilst retaining expressiveness.

In the current implementation of the constraint diagram editor [9] we choose a reading tree based on the order a user drew the spiders (or quantifiers) in the diagram.

If the spatial layout of spiders in a diagram is found to be important, then perhaps an editor could redraw diagrams to better match people's intuitions in general; for instance positioning the spider at which the user should start to read on the left. Similarly, such a layout could be used by an automatic proof writer when drawing diagrams and reading trees obtained by the application of some reasoning rules.

Notation

Due to lack of space we will not introduce the notation in detail (nor elucidate fully on the principles) but give a single example of a constraint diagram, without a reading tree, in figure 1; we refer the interested reader to [2,4,5].
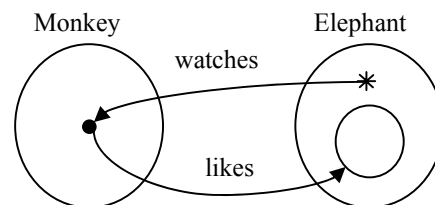

Figure 1. A constraint diagram without a reading tree.

In figure 1, there are two given contours labelled Monkey and Elephant which represent disjoint sets. The dot is an existential spider, which represents

existential quantification (there is a monkey). Its habitat is "inside Monkey". The asterisk is a universal spider, which represents universal quantification (for all elephants). The two arrows labeled "watches", and "likes" represent relations. The arrow labeled "likes" is sourced at the existential spider and targets an unlabelled derived contour inside Elephant. This derived contour represents the image of the relation "likes" (the elephants which are liked). The habitat of the universal spider is "inside Elephant outside the derived contour".

Two possible readings for figure 1 are:
  (a) There is a monkey, *m*, who likes only elephants and every elephant not liked by *m* watches just *m*.
  (b) Each elephant, *e*, watches a monkey who likes only elephants excluding *e*.

A spider represents a quantifier, and the domain of a spider represents the set over which quantification takes place. In reading (b), the domain of the universal spider is the whole of Elephant, while in reading (a) its domain is inside Elephant, but outside the derived contour.

Principles being tested

i. *Spider Type (or Quantifier Type)* - read existential spiders (dots) before universal ones (asterisks); proposed in [12] as a way to reduce ambiguity. Prediction for figure 1 is reading (a).

ii. *Following Chains of Arrows* - read from the start of a chain of arrows; proposed in [4,7]. A Chain of Arrows is a sequence of arrows where the source of an arrow is either equal to the target of the preceding arrow or is dependent upon it. For example in Figure 1, the pair (watches, likes) is a chain of arrows since the source of "likes" is equal to the target of "watches", and the pair (likes, watches) is a chain of arrows since the source of "watches" is inside Elephant, but outside the derived contour which is the target of "likes". No prediction for figure 1.

iii. *Strongly Versus Weakly Bound* – if there is more than one chain of arrows then read from the start of the one which has the target of each arrow equal to the source of the next (this is called strongly bound); proposed in [4]. Prediction for figure 1 is reading (b).

iv. *Domain Equals Habitat* – start reading at a spider whose domain is equal to its habitat; proposed in [4]. Prediction for figure 1 is (a).

v. *Outside a Derived Contour* – read from a spider which is not inside a derived contour; proposed in [5]. No prediction for figure 1.

vi. *The Positioning in the Plane of the Spiders* – read spiders from top to bottom, and from left to right; in most western languages, people read from top to bottom and from left to right. Reading from left to right predicts (a) for figure 1. Reading from top to bottom predicts (b).

Method

A mixture of a within-subjects and a between-subjects design was used. A within subjects design was used to investigate the principles: Spider Type, Following Chains of Arrows, Strongly Versus Weakly Bound, Domain Equals Habitat, and Outside a Derived Contour. A between subjects design was used to investigate the effect of the positioning in the plane of spiders. Forty subjects were assigned randomly to one (out of four) experimental condition. They were given a questionnaire to fill out, which started with an example-based explanation of the constraint diagram notation. It used only diagrams with one spider, so that there were no ambiguities or suggestions made here. Subjects were allowed to comment on the notation. Next, a sequence of Constraint diagrams was shown. For each diagram, two possible meanings were given, and subjects were asked to select the meaning that they thought the diagram represented most accurately, or to give an alternative meaning. The four conditions contained the same diagrams, except for layout changes, flipping the diagrams in the obvious manners.

Experimental Design – strengths

i. Keeping the diagrams as simple as possible whilst still testing the properties. Since ambiguous constraint diagrams (without a reading tree) have at least two spiders, at least one of which is universal, we chose examples which had exactly two spiders, at least one of which was universal.

ii. Labelling of diagrams used natural language, which removes alphabetic ordering interpretations (caused by labels like A, B, etc. on the contours).

iii. Labelling of diagrams used *unusual* natural language, which is not too semantically suggestive (using labels like "Teacher" and "Course" might have caused subjects to start with "Teacher" because they prefer starting with an actor rather than an object, for example; we chose labels such as "Monkey" and "Elephants" and "likes" and "watches").

iv. Various pairs of diagrams were used that differed in only one aspect, so that we could investigate the effect of most principles in isolation.

Experimental Design – weaknesses
 i. The order in which the answers are presented should have been randomized.
 ii. A larger number of participants would be beneficial.
 iii. Separable principles – some principles to be tested cannot be separated (perhaps this is an indication that for those principles another method of testing may be more appropriate).

Results
The only statistically significant results ($p < 0.05$ with Wilcoxon Signed Ranks Test) that were clearly interpretable were:
 i. If both of the properties of "domain equals habitat" and "following arrows" hold then there is a preference for starting reading at the predicted (by both of these properties) starting spider.
 ii. There was a preference for starting with the spider on the left.

Conclusions
The experiment shows how difficult evaluation of such complex notations can be. Despite running six pilot studies (each with two or more subjects) before the experiment, in order to try to optimize the experimental material (such as the explanation of the notation), we experienced problems with subjects not understanding everything (such as a two-footed spider).

A major problem of the experiment was that due to the inter-relationships between some properties, some diagrams represented multiple principles. We solved this problem to a large extent by comparing pairs of diagrams which differed in a minor way. However, some principles could not be separated by this approach. This made it impossible to decide if some of the non-significant results were caused by conflicting preferences for different principles. One option is to allow users of a constraint diagram editor to specify which principles they would like to use for a default reading. The layout of diagrams, appropriate highlighting, and the reading trees could then be altered accordingly.

The difficulties in isolating certain properties could point to a more qualitative method of investigation of user's preference (such as a thinking-aloud study, perhaps with eye-tracking facilities). It may also prove sensible to extensively test simpler notations, like Euler/Venn [13] and Spider diagrams [6] before testing Constraint diagrams.

Acknowledgements

References

 [1] Fish, A., and Flower, J., and Howse, J. A reading algorithm for constraint diagrams. Proceedings of VL/HCC 03, pp.161-168, IEEE, 2003.
 [2] Fish, A., and Flower, J., and Howse, J. The semantics of augmented constraint diagrams. To appear in Journal of Visual Languages and Computing, 2005.
 [3] Fish, A., and Howse, J. Computing reading trees for constraint diagrams. Proceedings of AGTIVE 03, pp.260-274., LNCS 3062, Springer, 2003.
 [4] Fish, A., and Howse, J. Towards a default reading for constraint diagrams. Proceedings of Diagrams '04, pp.51-65, LNAI 2980, Springer, 2004.
 [5] Fish, A., and Masthoff, J. (2005) Do Monkeys Like Elephants or Do Elephants Watch Monkeys? An Empirical study into the default reading of Constraint Diagrams. Technical report: http://cmis.mis.brighton.ac.uk/Research/vmg/publications.htm
 [6] Howse, J. Molina, F., Taylor, J., Kent, S. and Gil, J. Spider diagrams: A diagrammatic reasoning system. Journal of Visual Languages and Computing, vol 12, pp. 299-324, 2001.
 [7] Howse, J. and Schuman, S. Precise visual modelling: A case study. To appear in Software and Systems Modelling Journal, Vol.4, No.3, 2005.
 [8] Kent, S. Constraint diagrams: Visualising invariants in object oriented models. Proceedings of OOPSLA '97, ACM SIGPLAN Notices 32 Issue 10, pp. 327-341, 1997.
 [9] University of Kent's "Reasoning with Diagrams" site: http://www.cs.kent.ac.uk/projects/rwd/
 [10] OMG. UML Specification, Version 1.4. Available from www.omg.org.
 [11] Shin, S.-J. The logical status of diagrams. Cambridge University Press, 1994.
 [12] Stapleton, G., Howse, J, and Taylor, J. A constraint diagram reasoning system. Proceedings of the 9th International Conference on Distributed Multimedia Systems, International Conference on Visual Languages and Computing, pp. 263-270, Knowledge Systems Institute 2003.
 [13] Swoboda, N. and Allwein, G. Using DAG transformations to verify Euler/Venn homogeneous and Euler/Venn heterogeneous rules of inference. Journal of Software and System Modeling, Vol. 3, No.2, pp.136-149, 2004.
 [14] Warmer and Kleppe (1998). The Object Constraint Language. Addison-Wesley.