Aligning syntax and semantics in formalisations of visual languages

Corin Gurr
Division of Informatics, University of Edinburgh, Edinburgh EH8 9LW, UK
Email: C.Gurr@ed.ac.uk

Diagrams are popular, as many people find them more readily "accessible" than other forms of representation. Diagrams typically contain far more visible structure than any text-based representation and this structure can be used to reflect the structure of whatever it is that the diagram represents. Diagrams, however, are not are not always or automatically the most effective forms of representations. Furthermore, diagrams are typically not as expressive as text-based representations; the structures inherent in diagrams which make certain information more visible and inferences easier, also constrain what they are able to easily express. An effective diagram is typically taken to be one that is "well matched" to what it represents. This is to say, that the logical and spatio-visual properties of structures inherent to the diagram are chosen so as to have some very direct correspondence with the structures that they represent in the semantic domain; and in particular that they are chosen so as to support desired reasoning tasks by making certain inferences *immediate* and *obvious*.

Often the most effective diagrams are those which are very simple. However there is a strong tendency, particularly prevalent in visual formal modelling and specification languages, to take a diagrammatic language which at core is very simple, and then add many extensions and features to make it more expressive – often making it so expressive that the diagrams produced in the language are no longer readable; or at least, the diagrams are no longer obviously a more effective form of representation than a text-based one.

Both the design of effective visual formal modelling and specification languages, and the effective formalisation of (the semantics of) such visual languages requires the unification of results from visual language theory, cognitive science, empirical psychology and graphic design. Integrating results from such diverse fields is a non-trivial task, which may be approached through a decomposition of the study of issues of effectiveness in diagrammatic languages according to analogous understandings of (written and spoken) natural languages.

The study of natural languages is typically separated into the following categories: phonetics and phonology; morphology; syntax; semantics; pragmatics; and discourse. With the obvious exception of the first, the study of analogous categories in diagrammatic languages is at the same time both highly revealing of differences and similarities between the two forms of representation; and also provides a structure in which to explore the alternative means by which a diagram may capture meaning. Separating the study of diagrammatic languages into these categories permits us to firstly lay out the various means by which the structure inherent to diagrammatic morphologies and syntax may directly capture structure in the semantic domain; and secondly to consider how further pragmatic usage may convey meaning in diagrams. Such a study is undertaken in [1], in decomposing the variety of issues pertaining to effectiveness in diagrams.

Morphology concerns the shape of symbols. In contrast to typical textual languages, the basic vocabulary elements in some diagrammatic language may include such diverse shapes as circles, ellipses, squares, arcs and arrows. These objects often fall naturally into a hierarchy which can constrain the syntax and, furthermore, inform the semantics of the system. This hierarchy may be directly exploited by the semantics of symbols so as to reflect the depicted domain. In addition to a morphological partial typing, symbols may be further categorised through graphical properties such as size, colour, texture, shading and orientation. For example, the meaning of symbols represented by circles may be refined by distinguishing between large and small, and different coloured circles. Thus, again, part of the structure in the semantic domain is directly captured by morphological or syntactic features. The primary properties of graphical symbols may be considered to be: value (greyscale shading); texture (patterns); colour; orientation; size; thickness.

When we consider the use of spatio-visual relationships in the syntax of diagrammatic languages, we may see even stronger uses of "directness" to match the structure of a diagram to its meaning. The classic example of this is the representation of transitivity by a graphical relation such as spatial inclusion. A primary argument put forward to justify the claim of diagrammatic representation systems being more effective than textual ones is that certain inferences are somehow more immediate, or even are automatic, in dia-

Proceedings of the IEEE Symposia on Human-Centric Computing Languages and Environments (HCC'01)
0-7695-0474-4/01 $17.00 © 2001 IEEE

grams. In such representational systems conclusions appear "for free", as compared with textual systems where an inference must be made to produce the conclusion. Such phenomena have been termed inferential "free-rides", although studies such as [1] have shown the issue to be rather more complex and suggest that it would seem more accurate to term such occurrences "cheap rides" rather than free rides, with the addendum that some rides are cheaper than others.

An example of a "cheap ride" involves the use of a spatial dimension to represent some semantic notion. One common example is the representation of time as either the horizontal or vertical axis of a diagram. In UML sequence diagrams, for example, the vertical dimension represents time, flowing downwards in this case. For each object in the sequence diagram, represented as a labelled box, the vertical line below represents its lifeline and is widened when the object is active, dashed when it is inactive.

This use of a spatial dimension to represent time is common and the intrinsic properties are well matched (continuous, uni-dimensional, directional). It also has the advantage that while a particular moment in time can be readily identified yet all time is represented. Thus, for example, the diagram is easily searched for a particular moment, the ordering between two moments in time is obvious, etc. The disadvantage of representing time in this way is that one of the spatial dimensions is now "reserved", thus 2-dimensional elements and 2-dimensional layout cannot be utilised without leading to potential confusion. In UML sequence diagrams the structure of any system being represented is very simple: an actor and some number of associated objects with no representation of the relationship between these. This structure is necessarily simple, as a more complex layout would require the use of the vertical dimension, potentially confusing structure with timing. A common solution to this problem is to use multiple diagrams to represent a system, thus separating static structure from timing information. UML has class diagrams for represented static structure and other diagrams – such as the sequence diagrams above – for representing dynamic interactions. Once we accept that multiple diagrammatic languages in combination are required to represent some system, issues arise of consistency between diagrams.

Good use of layout and positioning is clearly a significant factor in determining the effectiveness of a diagram; although there are some difficulties, as morphological properties such as size, and syntactic relations such as *contains* or *overlaps*, affect layout and thus are potentially semantically affected whenever layout is directly interpreted. In many diagrammatic languages, such as graphs (i.e. "pure" node and link representations), layout is not explicitly semantically interpreted; yet even then its informal usage to convey information is common.

In linguistic theories of human communication, developed initially for written text or spoken dialogues, theories of pragmatics seek to explain how conventions and patterns of language use carry information over and above the literal truth value of sentences. Pragmatics, thus, helps to bridge the gap between truth conditions and "real" meaning - that is, between what is *said* and what is *meant*. This concept applies equally well to diagrams. Indeed, there is a recent history of work which draws parallels between pragmatic phenomena which occur in natural language, and for which there are established theories, and phenomena occurring in visual languages.

The use of layout in a pragmatic manner - that is, to capture meaning which is not necessarily readily expressed in the "official" semantics for a diagram - is extremely common. For example, studies of digital electronics engineers using CAD systems for designing the layout of computer circuits demonstrated that the most significant difference between novices and experts is in the use of layout to capture domain information. In such circuit diagrams the layout of components is not specified as being semantically significant. Nevertheless, experienced designers exploit layout to carry important information by grouping together components which are functionally related. By contrast, certain diagrams produced by novices were considered poor because they either failed to use layout or, in particularly "awful"' examples, were especially confusing through their misuse of the common layout conventions adopted by the experienced engineers. More recent studies of the users of various other visual languages, notably visual programming languages, have highlighted similar usage of graphical pragmatics. A major conclusion of this collection of studies is that the correct use of pragmatic features, such as layout in node and link based notations, is a significant contributory factor in the comprehensibility, and hence usability, of these representations.

Note that such use of layout is but one example of graphical pragmatics, albeit a common one. Essentially, any unexploited graphical feature (morphological hierarchies, colour, size, unused spatial relations, etc) may be used pragmatically to convey information without disturbing the semantics of a diagram. The potential for such pragmatic usage should be viewed as a bonus – or even a requirement – of a diagrammatic language. However, the construction of any specific diagram must also ensure that any non-semantic aspects are normalised as far as possible, as random or careless use of colour or layout, for example, can lead to unwanted mis- or over-interpretation by the reader.

References

[1] C. A. Gurr. Effective diagrammatic communication: Syntactic, semantic and pragmatic issues. *Journal of Visual Languages and Computing*, 10(4):317–342, Aug. 1999.

0-7695-0474-4/01 $17.00 © 2001 IEEE